Multi-output Local Gaussian Process Regression: Applications to Uncertainty Quantification

Ilias Bilionis[a,b], Nicholas Zabaras[b,a,*]

[a] *Center for Applied Mathematics, Cornell University*
[b] *Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, 101 Frank H. T. Rhodes Hall, Cornell University, Ithaca, NY 14853-3801, USA*

Abstract

We develop an efficient, Bayesian Uncertainty Quantification framework using a novel treed Gaussian process model. The tree is adaptively constructed using information conveyed by the observed data about the length scales of the underlying process. On each leaf of the tree, we utilize Bayesian Experimental Design techniques in order to learn a multi-output Gaussian process. The constructed surrogate can provide analytical point estimates, as well as error bars, for the statistics of interest. We numerically demonstrate the effectiveness of the suggested framework in identifying discontinuities, local features and unimportant dimensions in the solution of stochastic differential equations.

Keywords: Gaussian Process, Bayesian, Uncertainty quantification, Stochastic partial differential equations, Multi-output, Multi-element, Adaptivity.

1. Introduction

Uncertainty Quantification (UQ) is a field of great importance in practically all engineering tasks. Physical models require as input certain parameters such as physical constants, equations of state, geometric specification of objects, boundary conditions, initial conditions and so on. In general, exact knowledge of these quantities is impossible either due to measurement

*Corresponding author: Fax: 607-255-1222, Email: zabaras@cornell.edu, URL: http://mpdc.mae.cornell.edu/

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
07 DEC 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Multi-output Local Gaussian Process Regression: Applications to Uncertainty Quantification	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Cornell University,Sibley School of Mechanical and Aerospace Engineering,Materials Process Design and Control Laboratory,Ithaca,NY,14853**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
We develop an efficient, Bayesian Uncertainty Quantification framework us- ing a novel treed Gaussian process model. The tree is adaptively constructed using information conveyed by the observed data about the length scales of the underlying process. On each leaf of the tree, we utilize Bayesian Experi- mental Design techniques in order to learn a multi-output Gaussian process. The constructed surrogate can provide analytical point estimates, as well as error bars, for the statistics of interest. We numerically demonstrate the effectiveness of the suggested framework in identifying discontinuities, local features and unimportant dimensions in the solution of stochastic differential equations.

15. SUBJECT TERMS
Gaussian Process, Bayesian, Uncertainty quantification, Stochastic partial differential equations, Multi-output, Multi-element, Adaptivity.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**62**	

errors or because they are truly random. As a consequence, both the input parameters as well as the physical responses have to be modeled as random variables. The goal of UQ is to study the propagation of uncertainty from the parameter space to the response space. The most celebrated method for the solution of UQ problems is the Monte Carlo (MC) method. MC's wide acceptance is due to the fact that it can uncover the complete statistics of the solution, while having a convergence rate that is (remarkably) independent of the input dimension. Nevertheless, it quickly becomes inefficient in high dimensional and computationally intensive problems, where only a few samples can be observed. Such difficulties have been (partially) alleviated by improved sampling techniques such as Latin hypercube sampling [1] and multilevel MC [2, 3].

Another approach to UQ is the so called *spectral finite element method* [4]. It involves the projection of the response on a space spanned by orthogonal polynomials of the random variables and the solution of a system of coupled deterministic equations involving the coefficients of these polynomials. The scheme was originally developed for Gaussian random variables which correspond to Hermite polynomials (polynomial chaos (PC)). It was later generalized to include other types of random variables (generalized PC (gPC)) [5]. Due to the global support of the polynomials used, gPC suffers from the well-known Gibbs phenomenon in the presence of discontinuities in the random space. The multi-element generalized polynomial chaos (ME-gPC) method [6, 7] was introduced in order to address exactly this issue. The idea of the multi-element (ME) approach is to decompose the stochastic space in disjoint elements and then employ gPC on each element. However, the coupled nature of the equations that determine the coefficients of the polynomials make the application of the method to high input dimensions extremely difficult (*curse of dimensionality*).

Throughout the paper, we assume that we have at hand a well-established computer code that emulates the physical system. In fact, we will investigate the propagation of uncertainty from the input of the computer code to the output, by learning the response surface using well selected observations. Any modeling or discretization error will be ignored in this study. The so called *stochastic collocation* methods have been designed to deal with this situation. The response is represented as an interpolative polynomial of the random input constructed by calls to the computer code at specific input points. However, the construction of the set of interpolation points is non-trivial, especially in high-dimensional settings. In [8], a Galerkin based approximation

was introduced in conjunction with a collocation scheme based on a tensor product rule using one-dimensional Gauss quadrature points. Despite its appeal, the method scales badly with the number of random input dimensions. Alternatively, sparse grids (SG) based on the Smolyak algorithm [9] have a weaker dependence on the input dimensionality. In [10, 11, 12], the Smolyak algorithm is employed to build sparse grid interpolants in high-dimensional input spaces based on Lagrange interpolation polynomials. Similarly to gPC, such methods also fail to capture local features of the response. From the above discussion, it is apparent that discontinuities in the stochastic space must be dealt with using a basis with local support. In [13], the authors developed an adaptive version of SG collocation (SGC) based on localized hat functions called Adaptive SGC (ASGC). ASGC is able to refine the sparse grid only in important regions of the stochastic space, e.g. near a discontinuity. Nevertheless, the piecewise linear nature of the scheme performs poorly when only a few samples are used while adverse functions can trick the adaptive algorithm into stopping without converging.

Highly sophisticated computer codes modeling real-life phenomena (like weather, ocean waves, earthquakes, etc.) might take hours or even days to complete a single run in massively parallel systems. Therefore, we are necessarily limited to observing only a few realizations. Motivated by this situation, we would like to consider the problems of (1) selecting the most informative observations and (2) quantifying the uncertainty in the prediction of the statistics. From the above mentioned methods, ASGC addresses only problem (1), albeit in an ad hoc manner. In order to deal with (1) and (2) in a principled, information theoretic way, a Bayesian framework is necessary. To this end, we choose to investigate the performance of the Gaussian process (GP) model. The GP model has been used in computer experiments in the pioneering work of Sacks [14] (for a more recent review see the book [15]). GP is particularly interesting, since it provides an analytically tractable Bayesian framework where prior information about the response surface can be encoded in the covariance function, and the uncertainty about the prediction is easily quantified. It is exactly this uncertainty in the prediction that can be exploited in order to select the observations to be made (see [16]), as well as to quantify the uncertainty in the statistics. One of the drawbacks of GP inference is that it scales as the cube of the number of observations, making the treatment of large data sets computationally demanding. Furthermore, the most common covariance functions used in practice are stationary. The effect of the stationarity assumption is

that it makes non-stationary responses and localized features (such as discontinuities) a priori highly improbable, resulting in an excessive number of samples being required in order to uncover them. A successful effort to deal with these difficulties has been carried out in [17]. Based on the partitioning ideas of the Bayesian CART model [18, 19], a treed GP model was introduced. By making the GP local to each leaf of the tree, the model is able to process many more samples. Additionally, anisotropy is captured by considering the true response as being the result of many local stationary (albeit different) models. More recently, in [20] a novel tree model was introduced using Sequential Monte Carlo inference as opposed to MCMC of the classical approaches. The latter is a promising step towards computationally tractable fully Bayesian trees.

In this work, we present a novel non-intrusive UQ framework based on a treed multi-output Gaussian process (GP). It operates in two stages: (a) the construction of a surrogate model for the physical response and (b) the interrogation of this surrogate for the statistics. The building block of the surrogate is a Multi-output Gaussian Process (MGP) introduced in Section 2.1. Information gathered from the MGP is used to discover important directions of the stochastic space and decompose it in *stochastic elements* (i.e. new leaves of the tree) (Section 2.4). Each stochastic element is, in turn, sampled using Sequential Experimental Design (SED) techniques (Section 2.5) and subsequently modeled using a new MGP. This defines an iterative procedure that gradually resolves local features and discontinuities. The final result is a piecewise surrogate in the spirit of the Multi-element Method (ME) [6]. Despite being a treed GP, our model differs from the model in [17] in several aspects: 1) the tree building process is inspired from the ME method rather than Bayesian CART (non-probabilistic tree), 2) we explicitly derive point estimates of the missing hyper-parameters by maximizing the marginal likelihood instead of averaging (fast predictions), 3) we treat the multiple outputs of the response in a unified way (faster training). Furthermore, our model is built specifically to deal with UQ tasks, in that the input probability distribution plays an important role in the tree construction. Finally, the resulting surrogate can be used to obtain semi-analytic estimates of the moments of any order as well as error bars (Sections 2.2 and 2.3).

2. Methodology

Let $\mathbf{X} \subset \mathbb{R}^K$ for some $K \geq 1$ represent the stochastic input space, a (potentially infinite) rectangle of $\mathbb{R}^K$, i.e. $\mathbf{X} = \times_{k=1}^{K}[a_k, b_k], -\infty \leq a_k < b_k \leq \infty$. We will assume that there is a probability density $p(\mathbf{x})$ defined for all $\mathbf{x} \in \mathbf{X}$ such that:

$$p(\mathbf{x}) = \prod_{k=1}^{K} p_k(x_k), \tag{1}$$

where $p_k(x_k)$ is the probability density pertaining to the k-th dimension. That is, the components of $\mathbf{x}$ are independent random variables. This assumption is very common in UQ settings and can be made to hold by a transformation of the input space. We now consider the multi-output function $\mathbf{f} : \mathbf{X} \to \mathbb{R}^M$ representing the result of a computer code (deterministic solver) modeling a physical system, i.e. at a given input point $\mathbf{x} \in \mathbf{X}$ the response of the system is $\mathbf{f}(\mathbf{x})$. We will write

$$\mathbf{f}(\cdot) = (f_1(\cdot), \ldots, f_M(\cdot)),$$

and refer to $f_r(\cdot)$ as the r-th output of the response function, $r = 1, \ldots, M$. In this work, we will identify $\mathbf{f}(\cdot)$ as the true response of an underlying physical system and we will ignore any modeling errors. The input probability distribution induces a probability distribution on the output. The UQ problem involves the calculation of the statistics of the output $\mathbf{y} = \mathbf{f}(\mathbf{x})$. Quantities of interest are the *moments* $\mathbf{m}^q = (m_1^q, \ldots, m_M^q)$, defined for $q \geq 1$ and $r = 1, \ldots, M$ by:

$$m_r^q := \int_{\mathbf{X}} f_r^q(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}, \tag{2}$$

as well as functions of them. In particular, the *mean* $\mathbf{m} = (m_1, \ldots, m_M)$:

$$m_r := m_r^1 = \int_{\mathbf{X}} f_r(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}, \tag{3}$$

and the *variance* $\mathbf{v} = (v_1, \ldots, v_M)$:

$$v_r := \int_{\mathbf{X}} \left(f_r(\mathbf{x}) - m_r\right)^2 p(\mathbf{x}) d\mathbf{x} = m_r^2 - (m_r^1)^2. \tag{4}$$

The statistics will be calculated by interrogating a surrogate of $\mathbf{f} : \mathbf{X} \to \mathbb{R}^M$. This will be put together from local surrogates defined over elements of the stochastic space $\mathbf{X}^i \subset \mathbf{X}$ such that:

$$\mathbf{X} = \cup_{i=1}^{I} \mathbf{X}^i \text{ and } \operatorname{int}(\mathbf{X}^i) \cap \operatorname{int}(\mathbf{X}^j) = \emptyset, \forall i, j \in I, i \neq j, \tag{5}$$

where $\text{int}(\mathbf{X}^i)$ denotes the interior of the set $\mathbf{X}^i$ under the usual Euclidean metric of $\mathbb{R}^K$. The response surface is correspondingly decomposed as:

$$\mathbf{f}(\mathbf{x}) := \sum_{i=1}^{I} \mathbf{f}^i(\mathbf{x}) 1_{\mathbf{X}^i}(\mathbf{x}), \tag{6}$$

where $1_{\mathbf{X}^i}(\mathbf{x})$ is the indicator function of $\mathbf{X}^i$, given by:

$$1_{\mathbf{X}^i}(x) = \begin{cases} 1 & \text{if } \mathbf{x} \in \mathbf{X}^i, \\ 0 & \text{otherwise} \end{cases},$$

and $\mathbf{f}^i(\cdot)$ is just the restriction of $\mathbf{f}(\cdot)$ on $\mathbf{X}^i$. The local surrogates will be identified as Multi-Output Gaussian Processes (MGP) defined over the stochastic element $\mathbf{X}^i$. These MGPs will be trained by observing $\mathbf{f}^i(\cdot)$. The predictive mean of the MGPs will be used to derive semi-analytic estimates of all moments $\mathbf{m}^q$. An addendum of the Bayesian treatment, is the ability to provide error bars for the point estimates of the moments. This feature is absent from most current UQ methods.

Our aim is to create a surrogate by making as few calls to the computer program as possible. This is achieved by an interplay of adaptively decomposing the domain (Tree Construction) and selecting which observations to make within each element (Experimental Design). These decisions should be biased by the underlying input probability density $p(\mathbf{x})$ and the observed variability of the responses.

In the sections that follow, we introduce the constituent parts of our framework. Despite the fact that the method is applicable to any distribution $p(\mathbf{x})$ over $\mathbf{X}$, all numerical examples will be conducted on a compact $\mathbf{X}$ (a_k and b_k are finite) using the uniform distribution. This is mainly due to the fact that the implementation of the framework is considerably easier for this case. We plan to investigate and report the dependence of the results on $p(\mathbf{x})$ in a future work.

2.1. Multi-output Gaussian Process Regression

We turn our focus to a single element of the stochastic space $\mathbf{X}^i \subset \mathbf{X}$ and discuss the construction of a local surrogate model based on some already observed data. The choice of the elements is the subject of Section 2.4 and how the observations are selected is investigated in Section 2.5. All quantities introduced herein are local to the element $\mathbf{X}^i$. However, in order to avoid

having an unnecessarily complicated notation, we do not explicitly show this dependence.

We assume that we have observed a fixed number $N \geq 1$ of data points

$$\mathcal{D} := \left\{ (\mathbf{x}^{(n)}, \mathbf{y}^{(n)} \right\}_{n=1}^{N}, \tag{7}$$

where, $\mathbf{y}^{(n)} = \mathbf{f}(\mathbf{x}^{(n)})$ is the result of the computer program with input $\mathbf{x}^{(n)}$. We will fit these data to a Gaussian Process (GP) model [21, 22], a procedure known as GP Regression (GPR). Our primary concern in this section is to extend GPR to the multi-output case. The naive approach would be to model its output dimension independently. However, since the various outputs of the response function are highly correlated, this strategy will incur some loss of information. Furthermore, training a GP on N data points involves the computation of the Cholesky decomposition of an $N \times N$ symmetric positive-definite matrix, an operation that scales as $O(N^3)$. If the M output dimensions were to be modeled independently, then the total training cost would be $O(MN^3)$ making the method inappropriate for most UQ tasks. Several techniques exist that model the correlation between outputs: e.g. 'co-kriging' (Section 3.2.3 in [23]) or introducing latent (hidden) outputs [24, 25, 26]. Unfortunately, these models are still fairly complicated and computationally demanding. In [27], a principal components analysis (PCA) was performed on the output space and then the PCA coefficients of the simulations were modeled using independent GPs. This approach has been proven efficient in dealing with high-dimensional output settings, since it automatically takes care of output correlations. However, it introduces an additional error arising from the finite truncation of the PCA decomposition of the output field. Furthermore, it is not clear how the approach can be used in a SED setting, in which simulations arrive one by one, as well as how it performs when discontinuities are present in the stochastic space. A very recent, theoretically sound way of modeling multiple outputs was developed in [28]. In this approach, the multidimensional response is modeled as a GP vector using the same covariance function for each dimension. It accounts for correlations by introducing a constant correlation matrix between the outputs. However, in very high-dimensional settings (typical UQ applications have a few thousand outputs), dealing with the full correlation matrix is computationally challenging. Since in this work we are trying to develop a method that will be able to deal with output dimensions that range from a few hundreds to a few thousands, keeping the training time to acceptable

levels is one of our major goals. We achieve this by making a compromise: the outputs will be treated as *conditionally independent* given the covariance function. Our approach is similar to that in [28] if a diagonal correlation matrix and a constant mean is used. The underlying assumption is that the regularity of all output dimensions is approximately the same. Since each output may vary in signal strength (e.g. finite element nodes close to a fixed boundary condition exhibit smaller variations compared to ones in the middle of the domain), we have to work with a scaled version of the responses. The computational savings of using a single covariance function for all outputs are tremendous: only a single Cholesky decomposition is required, dropping the training cost back to $O(N^3)$. We call the resulting model a Multi-output Gaussian Process (MGP) and refer to regression using MGPs as MGPR.

Let us introduce the *observed mean*:

$$\mu_{\text{obs},r} = \frac{1}{N}\sum_{n=1}^{N} y_r^{(n)}, \tag{8}$$

and the *observed variance*:

$$\sigma^2_{\text{obs},r} = \frac{1}{N}\sum_{n=1}^{N} (y_r - \mu_{\text{obs},r})^2, \tag{9}$$

of the data $\mathcal{D}$. We will be modeling the *scaled response functions* $g_r : \mathbf{X}^i \to \mathbb{R}$, defined by:

$$g_r(\mathbf{x}) = \frac{f_r(\mathbf{x}) - \mu_{\text{obs},r}}{\sigma_{\text{obs},r}}, r = 1, \ldots, M. \tag{10}$$

The scaling is necessary, because the various outputs might exhibit different signal strengths. Obviously, this definition depends on the actual observations. We expect, however, that if N is big or if the stochastic element under investigation is small, then it is a good approximation to the ideal scaling, i.e. zero mean and unit variance for all outputs. Assuming that all outputs have the same regularity, we model each g_r as a Gaussian Process with zero mean and covariance function $c(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$:

$$g_r(\mathbf{x})|\boldsymbol{\theta} \sim \mathcal{GP}\left(0, c(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})\right), r = 1, \ldots, M,$$

where $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^S$ are the $S \geq 1$, unknown *hyper-parameters* of the covariance function. That is, the scaled responses are treated as conditionally independent given the hyper-parameters.

Point Estimates of the Hyper-parameters. A fully Bayesian approach would proceed by imposing a prior probability $\pi(\boldsymbol{\theta})$ over the hyper-parameters and then average (numerically) over them. Instead, we will employ the *evidence approximation* to Bayesian inference [29], in order to obtain point-estimates of the hyper-parameters by maximizing the marginal likelihood of the data (Ch. 5 of [22]). This necessarily underestimates the prediction uncertainty, but it is a trade-off we are willing to make in order to obtain a computationally tractable model. The logarithm of the marginal likelihood of each scaled response $g_r(\cdot), r = 1, \ldots, M$ is given by:

$$\log p(\mathbf{z}_r|\mathcal{D}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{z}_r^T\mathbf{C}^{-1}\mathbf{z}_r - \frac{1}{2}\log|\mathbf{C}| - \frac{N}{2}\log 2\pi,$$

where $\mathbf{z}_r = (z_r^{(1)}, \ldots, z_r^{(N)})$ is a scaled version of the observations in $\mathcal{D}$:

$$z_r^{(n)} = \frac{y_r^{(n)} - \mu_{\text{obs},r}}{\sigma_{\text{obs},r}}, n = 1, \ldots, N, \tag{11}$$

$\mathbf{C} = (C_{ij}), C_{ij} = c(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \boldsymbol{\theta})$ is the covariance matrix and $|\mathbf{C}|$ its determinant. Since the scaled responses are conditionally independent given $\boldsymbol{\theta}$, the logarithm of the joint marginal likelihood is simply the sum of the marginal likelihoods of each output, i.e.

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &:= \log p(\mathbf{z}_1, \ldots, \mathbf{z}_M|\mathbf{X}, \boldsymbol{\theta}) \\
&= \sum_{r=1}^{M} \log p(\mathbf{z}_r|\mathbf{X}, \boldsymbol{\theta}) \\
&= -\frac{1}{2}\sum_{r=1}^{M} \mathbf{z}_r^T\mathbf{C}^{-1}\mathbf{z}_r - \frac{M}{2}\log|\mathbf{C}| - \frac{NM}{2}\log 2\pi.
\end{aligned}$$

Thus, a point estimate of $\boldsymbol{\theta}$ over the element $\mathbf{X}^i$ is obtained by

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}). \tag{12}$$

The joint marginal likelihood $\mathcal{L}(\boldsymbol{\theta})$ might exhibit multiple maxima which correspond to alternative interpretations of the data. In practice, we make an educated initial guess and we are satisfied with the (local) maximum obtained using a Conjugate Gradient method [30]. The specifics of the optimization method are discussed in Appendix A.

The Predictive Distribution. Having decided on a point estimate for the hyper-parameters $\boldsymbol{\theta}$, we are ready to predict the scaled response at any test point $\mathbf{x} \in \mathbf{X}^i$. Scaling back to the original responses, we can easily see that the predictive distribution of $f_r(\mathbf{x})$ is:

$$f_r(\mathbf{x})|\mathcal{D}, \boldsymbol{\theta}^* \sim \mathcal{N}\left(\mu_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*), \sigma^2_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*)\right), \tag{13}$$

with mean:

$$\mu_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*) = \sigma_{\text{obs},r}\mathbf{c}^T\mathbf{C}^{-1}\mathbf{z}_r + \mu_{\text{obs},r}, \tag{14}$$

and variance:

$$\sigma^2_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*) = \sigma^2_{\text{obs},r}\left(c(\mathbf{x}, \mathbf{x}; \boldsymbol{\theta}^*) - \mathbf{c}^T\mathbf{C}^{-1}\mathbf{c}\right), \tag{15}$$

where $\mathbf{c} = (c(\mathbf{x}, \mathbf{x}^{(1)}; \boldsymbol{\theta}^*), \ldots, c(\mathbf{x}, \mathbf{x}^{(N)}; \boldsymbol{\theta}^*))$ and the covariance matrix $\mathbf{C}$ is evaluated at $\boldsymbol{\theta}^*$. We will refer to $\sigma^2_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*)$ as the *predictive variance* of the response at $\mathbf{x}$. It represents our uncertainty about the prediction at this particular test point.

As mentioned earlier, the *predictive mean* $\mu_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*)$ given by Eq. (14) will be used to provide estimates for the statistics over the element $\mathbf{X}^i$, while the predictive variance $\sigma^2_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*)$ will give error bars (see Section 2.2). Notice that $\mu_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*)$ is, in fact, a *kernel estimator* since:

$$\mu_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*) = \sum_{n=1}^{N} \alpha_{rn} c(\mathbf{x}^{(n)}, \mathbf{x}; \boldsymbol{\theta}^*) + \mu_{\text{obs},r}, \tag{16}$$

where the weights α_{rn} are given by:

$$\boldsymbol{\alpha}_r \equiv (\alpha_{r1}, \alpha_{r2}, \ldots, \alpha_{rN}) := \sigma_{\text{obs},r}\mathbf{C}^{-1}\mathbf{z}_r,$$

and also depend on $\boldsymbol{\theta}^*$ through $\mathbf{C}$.

2.2. Calculation of the local statistics

As in the previous section, we focus on a specific element $\mathbf{X}^i$. All quantities are again local to $\mathbf{X}^i$. In order to keep notational complexity to a minimum, we do not explicitly show this dependence. We will derive *analytic* point estimates as well as error bars for the mean and the higher moments of the response based on the linear point estimator of $\mathbf{f}(\cdot)$ over $\mathbf{X}^i$

given in Eq. (16) and the predictive variance Eq. (15). To be exact, we are interested in estimating all moments $\mathbf{m}^q = (m_1^q, \ldots, m_M^q), q \geq 1$, where

$$m_r^q = \int_{\mathbf{X}^i} f_r^q(\mathbf{x}) p^i(\mathbf{x}) d\mathbf{x}. \tag{17}$$

$p^i : \mathbf{X} \to \mathbb{R}$ is the *conditional probability density* related to $\mathbf{X}^i$:

$$p^i(\mathbf{x}) := \frac{p(\mathbf{x})}{P(\mathbf{X}^i)} 1_{\mathbf{X}^i}(\mathbf{x}), \tag{18}$$

where $P(\mathbf{X}^i)$ is the probability of an input point residing in the stochastic element $\mathbf{X}^i$, i.e.

$$P(\mathbf{X}^i) = \int_{\mathbf{X}^i} p(\mathbf{x}) d\mathbf{x}.$$

In order to achieve analytic estimates of $\mathbf{m}^q$, we keep concurrent MGP estimates of the response raised to the q power. In particular, the q power of the response is treated also as a MGP with its own hyper-parameters $\boldsymbol{\theta}^q$. Let us denote the predictive distribution for the q power of the response at $\mathbf{x} \in \mathbf{X}^i$ by:

$$f_r^q(\mathbf{x}) | \mathcal{D}, \boldsymbol{\theta}^q \sim \mathcal{N}\left(\mu_{f_r^q}(\mathbf{x}; \boldsymbol{\theta}^q), \sigma^2_{f_r^q}(\mathbf{x}; \boldsymbol{\theta}^q)\right),$$

where $\mu_{f_r^q}(\mathbf{x}; \boldsymbol{\theta}^q)$ is the predictive mean and $\sigma^2_{f_r^q}(\mathbf{x}; \boldsymbol{\theta}^q)$ the predictive variance for $r = 1, \ldots, M$. These quantities are available through the exact same procedure described in Section 2.1, using the q power of the response instead of the response itself. For convenience, let us write the predictive mean at $\mathbf{x}$ as:

$$\mu_{f_r^q}(\mathbf{x}; \boldsymbol{\theta}^q) = \sum_{n=1}^{N} \alpha_{rn}^q c(\mathbf{x}^{(n)}, \mathbf{x}; \boldsymbol{\theta}^q) + \mu_{\text{obs},r}^q,$$

and the predictive variance at $\mathbf{x}$ as:

$$\sigma^2_{f_r^q}(\mathbf{x}; \boldsymbol{\theta}^q) = (\sigma^q_{\text{obs},r})^2 \left(c(\mathbf{x}, \mathbf{x}; \boldsymbol{\theta}^q) - \mathbf{c}^{q,T} (\mathbf{C}^q)^{-1} \mathbf{c}^q\right),$$

where $\mu^q_{\text{obs},r}$ and $\sigma^q_{\text{obs},r}$ are defined as in Eqs. (8) and (9), respectively, using the q power of the observed response, $\mathbf{c}^q = (c(\mathbf{x}^{(1)}, \mathbf{x}; \boldsymbol{\theta}^q), \ldots, c(\mathbf{x}^{(N)}, \mathbf{x}; \boldsymbol{\theta}^q))$ and $\mathbf{C}^q$ is the covariance matrix evaluated at $\boldsymbol{\theta}^q$.

Our goal is to derive a predictive probability distribution for the moments $\mathbf{m}^q$ given the data and the hyper-parameters. In a proper probabilistic treatment, we would proceed by sampling the full posterior of the MGP,

integrating the samples over $\mathbf{x}$ and producing a Monte Carlo estimate of the predictive mean and variance of each moment. To obtain analytic estimates, let us make the simplifying assumption that the predictions at different input points $\mathbf{x}$ are conditionally independent given the data and the hyperparameters. Then, by the additivity of independent normal variables, we arrive at the approximation:

$$m_r^q|\mathcal{D}, \boldsymbol{\theta}^q \sim \mathcal{N}\left(\mu_{m_r^q}, \sigma^2_{m_r^q}\right), \tag{19}$$

where the predictive mean of m_r^q is:

$$\mu_{m_r^q} = \int_{\mathbf{X}^i} \mu_{f_r^q}(\mathbf{x}; \boldsymbol{\theta}^q) p^i(\mathbf{x}) d\mathbf{x}, \tag{20}$$

and its predictive variance:

$$\sigma^2_{m_r^q} = \int_{\mathbf{X}^i} \sigma^2_{f_r^q}(\mathbf{x}; \boldsymbol{\theta}^q) p^i(\mathbf{x}) d\mathbf{x}. \tag{21}$$

Fortunately, the integrals involved can be expressed in terms of expectations of the covariance function with respect to the conditional input distribution. This results in a fast, semi-analytic estimate of $\mu_{m_r^q}$ and $\sigma_{m_r^q}$. It is worth mentioning at this point that this distribution is necessarily wider than the optimum one.

Remark 1. Obviously, the assumption that a positive function, e.g. the response f_r raised to an even power, is a Gaussian Process is not optimal, since the predictive distribution assigns positive probability to the event of the function getting negative values. However, this assumption is necessary in order to obtain analytic estimates of the predictive distribution of the statistics. A direct consequence of it is that the predictive distribution Eq. (19) for an even moment has also positive probability of being negative. A tighter predictive distribution can always be found by truncating Eq. (21) below zero. On the other hand, the predictive mean of an even moment will always be positive.

Evaluation of the integrals. We now proceed to the calculation of the integrals in Eqs. (20) and (21). We can write the following:

$$\mu_{m_r^q} = \sum_{n=1}^{N} \alpha_{rn}^q \epsilon_n^q + \mu_r^q, \tag{22}$$

and

$$\sigma^2_{m_r^q} = (\sigma^q_{\text{obs},r})^2 \left(c^q - \sum_{n,l=1}^{N} (\mathbf{C}^q)^{-1}_{nl} \nu^q_{nl} \right), \tag{23}$$

where

$$\epsilon^q_n = \int_{\mathbf{X}^i} c(\mathbf{x}^{(n)}, \mathbf{x}; \boldsymbol{\theta}^q) p^i(\mathbf{x}) d\mathbf{x}, \tag{24}$$

$$c^q = \int_{\mathbf{X}^i} c(\mathbf{x}, \mathbf{x}; \boldsymbol{\theta}^q) p^i(\mathbf{x}) d\mathbf{x}, \tag{25}$$

$$\nu^q_{nm} = \int_{\mathbf{X}^i} c(\mathbf{x}, \mathbf{x}^{(n)}; \boldsymbol{\theta}^q) c(\mathbf{x}, \mathbf{x}^{(l)}; \boldsymbol{\theta}^q) p^i(\mathbf{x}) d\mathbf{x}, \tag{26}$$

and $(\mathbf{C}^q)^{-1}_{nl}$ is the nl element of the inverse q covariance matrix $(\mathbf{C}^q)^{-1}$.

Thus, computation of the statistics requires the evaluation of integrals of the form of Eqs. (24), (25) and (26). In Appendix A, we provide analytic formulas for their calculation for the special case of uniform input distribution and Squared Exponential (SE) covariance function. For the SE covariance function but arbitrary input probability density of the form of Eq. (1), their evaluation requires $O(K)$ one-dimensional numerical integrations.

2.3. From local to global statistics

In the same spirit as the multi-element methods [6, 7, 31], we combine the statistics over each stochastic element in order to obtain their global analogues. Since we now work over the whole domain, we will explicitly mark the dependence of the underlying quantities on the element $\mathbf{X}^i, i = 1, \ldots I$. Let $m_r^{q,i}$ be the q moment of the response that pertains to the conditional probability density $p^i(\mathbf{x})$ (Eq. (17)) and m_r^q be the global one (Eq. (2)). Notice that m_r^q can be decomposed as

$$\begin{aligned} m_r^q &= \int_{\mathbf{X}} f_r^q(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \\ &= \sum_{i=1}^{I} \int_{\mathbf{X}^i} f_r^q(\mathbf{x}) \frac{p(\mathbf{x})}{P(\mathbf{X}^i)} d\mathbf{x} P(\mathbf{X}^i) \\ &= \sum_{i=1}^{I} \int_{\mathbf{X}^i} f_r^q(\mathbf{x}) p^i(\mathbf{x}) d\mathbf{x} P(\mathbf{X}^i), \end{aligned}$$

or

$$m_r^q = \sum_{i=1}^{I} m_r^{q,i} P(\mathbf{X}^i). \tag{27}$$

Now, assume that for each element $\mathbf{X}^i, i = 1, \ldots I$ we have obtained a predictive distribution (Eq. (19)) for $m_r^{q,i}$ and let its predictive mean and variance be $\mu_{m_r^{q,i}}$ and $(\sigma_{m_r^{q,i}})^2$, respectively (Eqs. (22) and (23)). Assuming conditional independence of the predictive distributions given the data and the hyper-parameters, we obtain that:

$$m_r^q | \mathcal{D}, \boldsymbol{\theta}^q \sim \mathcal{N}\left(\mu_{m_r^q}, \sigma^2_{m_r^q}\right), \tag{28}$$

where the predictive mean is:

$$\mu_{m_r^q} = \sum_{i=1}^{I} \mu_{m_r^{q,i}} P(\mathbf{X}^i), \tag{29}$$

and the predictive variance:

$$\sigma^2_{m_r^q} = \sum_{i=1}^{I} \sigma^2_{m_r^{q,i}} P(\mathbf{X}^i). \tag{30}$$

Again, truncation of this distribution below zero for even q, always yields an improved estimator (see Remark 1).

Finally, we derive a normal approximation to the predictive distribution for the variance of the response $\mathbf{v} = (v_1, \ldots, v_M)$ (defined in Eq. (4)):

$$v_r \sim \mathcal{N}\left(\mu_{v_r}, \sigma^2_{v_r}\right). \tag{31}$$

Under the assumption of conditional independence of $m_r^q,\ q = 1, 2$, the predictive mean of v_r is given by:

$$\begin{aligned} \mu_{v_r} &:= \mathbf{E}\left[m_r^2 - (m_r^1)^2 | \mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right] \\ &= \mathbf{E}\left[m_r^2 \middle| \mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right] - \mathbf{E}\left[(m_r^1)^2 | \mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right], \end{aligned}$$

or:

$$\mu_{v_r} = \mu_{m_r^2} - \mu^2_{m_r^1} - \sigma^2_{m_r^1}, \tag{32}$$

where $\mathbf{E}[\cdot|\mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2]$ denotes the expectation with respect to the joint predictive distribution for m_r^1 and m_r^2. Equivalently, the predictive variance is:

$$\begin{aligned}
\sigma_{v_r}^2 &:= \mathbf{V}\left[m_r^2 - (m_r^1)^2 | \mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right] \\
&= \mathbf{V}\left[m_r^2 | \mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right] + \mathbf{V}\left[(m_r^1)^2 | \mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right] \\
&= \mathbf{V}\left[m_r^2 | \mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right] + \mathbf{E}\left[(m_r^1)^4 | \mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right] - \left(\mathbf{E}\left[(m_r^1)^2 | \mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right]\right)^2,
\end{aligned}$$

or:

$$\sigma_{v_r}^2 = \sigma_{m_r^2}^2 + 4\mu_{m_r^1}^2 \sigma_{\mu_r^1}^2 + 2\sigma_{\mu_r^1}^4, \tag{33}$$

where $\mathbf{V}[\cdot|\mathcal{D}, \boldsymbol{\theta}^1, \boldsymbol{\theta}^2]$ denotes the variance with respect to the joint predictive distribution of m_r^1 and m_r^2.

Let us end this section by mentioning that the above procedure can be easily applied to obtain normal approximations to the predictive distributions of any centered moment. It is obvious that the calculation can always be casted in terms of moments of the normal distribution which are readily available using the *confluent hypergeometric function* $U(a, b, x)$ (see Ch. 13 of [32]).

2.4. Adaptivity

In this section, we develop an iterative procedure to adaptively decompose the stochastic space in smaller elements. The initial step of this procedure starts by considering a single element, i.e. $\mathbf{X}$ itself. Here, we assume that we are already given a decomposition of the domain as well as a local surrogate model on each element. The decision we wish to make is whether or not to refine a given element and in which way. We develop refinement criteria that are based solely on information gathered by the current surrogate model and no further calls to the deterministic solver are required. The Bayesian predictive variance Eq. (15) is used to define a measure of our uncertainty about the prediction over the whole domain $\mathbf{X}$. We show how this measure can be broken down to contributions coming from each element. Based on this observation, we derive a criterion that suggests refinement of an element if its contribution to the global uncertainty is larger than a pre-specified threshold. For the sake of simplicity, we only consider rectangular elements and refine them by splitting them perpendicular to the dimension of greatest importance in two pieces of equal probability. The importance of a particular

dimension is characterized by its length scale. The length scales are identified as the hyper-parameters of a SE covariance function.

Suppose that we have already a decomposition of the stochastic domain $\mathbf{X}$ in *rectangular* elements $\mathbf{X}^i$, e.g.

$$\mathbf{X}^i = [a_1^i, b_1^i] \times \cdots \times [a_K^i, b_K^i],$$

with $a_k^i < b_k^i, k = 1, \ldots, K, i = 1, \ldots, I$ such that Eq. (5) holds. Furthermore, assume that we have already learnt the local surrogates on each element $\mathbf{X}^i$. Let $\sigma^2_{f_r^i}(\mathbf{x})$ be the predictive variance of the $r = 1, \ldots, M$ output of the local surrogate of $\mathbf{f}^i$ at $\mathbf{x} \in \mathbf{X}^i$ (Eq. (15)). By the conditional independence assumption for the predictive distribution over each element and Eq. (6), the predictive variance of the $r = 1, \ldots, M$ dimension of the global surrogate $\sigma^2_{f_r}(\mathbf{x})$ at $\mathbf{x} \in \mathbf{X}$ is given by:

$$\sigma^2_{f_r}(\mathbf{x}) = \sum_{i=1}^{I} \sigma^2_{f_r^i}(\mathbf{x}) 1_{\mathbf{X}^i}(\mathbf{x}). \tag{34}$$

Its average over r,

$$\sigma^2_{\mathbf{f}}(\mathbf{x}) := \frac{1}{M} \sigma^2_{f_r^i}(\mathbf{x}),$$

is a measure of our uncertainty about the prediction of all outputs simultaneously at the test point $\mathbf{x} \in \mathbf{X}$. Taking the expectation of this quantity with respect to the input probability density $p(\mathbf{x})$, we obtain

$$\sigma^2_{\mathbf{f},p} := \int_{\mathbf{X}} \sigma^2_{\mathbf{f}}(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}. \tag{35}$$

This quantity is a measure of our uncertainty about our prediction over the whole domain $\mathbf{X}$. Notice that, in $\sigma^2_{\mathbf{f},p}$, the uncertainty of the model at $\mathbf{x}$ is weighted by its probability of occurrence $p(\mathbf{x})$. Intuitively speaking, we are willing to accept a somewhat less accurate surrogate in regions of the space occurring with lower probability. Using Eq. (34), it is straightforward to see that:

$$\sigma^2_{\mathbf{f},p} = \sum_{i=1}^{I} \sigma^2_{\mathbf{f},p^i} P(\mathbf{X}^i), \tag{36}$$

where

$$\sigma^2_{\mathbf{f},p^i} := \int_{\mathbf{X}^i} \sigma^2_{\mathbf{f}}(\mathbf{x}) p^i(\mathbf{x}) d\mathbf{x},$$

is the uncertainty of our prediction over the element $\mathbf{X}^i$. Making use of Eq. (21) for $q = 1$, we obtain that:

$$\sigma^2_{\mathbf{f},p^i} = \frac{1}{M} \sum_{r=1}^{M} \sigma^2_{m_r^{1,i}}. \tag{37}$$

Hence, $\sigma^2_{\mathbf{f},p^i}$ relates directly to our uncertainty about the mean response $\sigma^2_{m_r^{1,i}}$ (Eq. (23)). Generalizing, we can define the corresponding uncertainties for the response raised to the $q \geq 1$ power (see Section 2.3):

$$\sigma^2_{\mathbf{f}^q,p} := \sum_{i=1}^{I} \sigma^2_{\mathbf{f}^q,p^i} P(\mathbf{X}^i), \tag{38}$$

where

$$\sigma^2_{\mathbf{f}^q,p^i} := \frac{1}{M} \sum_{r=1}^{M} \sigma^2_{m_r^{q,i}}. \tag{39}$$

This measure is equivalent to our uncertainty about the q-th moment of the response. Our idea it to refine the element $\mathbf{X}^i$, if the contribution to the global uncertainty coming from it, is greater than a certain threshold $\delta > 0$, that is we refine $\mathbf{X}^i$ if:

$$\sigma^2_{\mathbf{f}^q,p^i} P(\mathbf{X}^i) > \delta, \text{ for any } q = 1, 2, \ldots, \tag{40}$$

depending on how many moments one wishes to consider. However, in the numerical examples of the present work, we simply use the criterion for $q = 1$, despite the fact that we report also the variance. We plan to investigate its dependence on q in a later work.

The above criterion specifies whether or not an element $\mathbf{X}^i$ should be refined. As already mentioned, we refine elements by cutting them in equally probable parts perpendicular to 'the most important dimension'. At this point, we attempt to give a precise meaning to the concept of 'the most important dimension'. Towards this goal, we will exploit the properties of a specific parametric form for the covariance function, the Squared Exponential (SE):

$$c_{\text{SE}}(\mathbf{x}, \mathbf{x}') = s_f^2 \exp\left(-\frac{1}{2} \sum_{k=1}^{K} \frac{(x_k - x'_k)^2}{\ell_k^2} \right), \tag{41}$$

where $s_f > 0$ can be interpreted as the *signal strength* and $\ell_k > 0$ as the *length scale* of each stochastic input. These parameters can be learnt from the data by using the evidence approximation (see Section 2.1), allowing the determination of the relative importance of each dimension. The technique is called *automatic relevance determination* (ARD). It originated in the Neural Networks literature [33] and was later extended to GP Regression [34]. We emphasize that a unique set of the SE hyper-parameters is learnt on each element $\mathbf{X}^i$ (as well as for each power of the response, $\mathbf{f}^q$, that we take into account). Hence, despite the fact that each local surrogate is a stationary GP, the global surrogate is non-stationary. This is similar in spirit to the Bayesian Treed Gaussian Process Model in [17].

Let us explicitly denote the learnt length scales of element $\mathbf{X}^i$ corresponding to the MGP that represents $\mathbf{f}$, with $\ell_k^i, k = 1, \ldots, K$. The length scales of the powers of the response, $\mathbf{f}^q, q > 1$, are not involved in the criterion we are about to formulate. Furthermore, let us introduce the probability P_k^i that the k-th dimension x_k of a random input point $\mathbf{x} \in \mathbf{X}$ falls inside $\mathbf{X}^i$:

$$P_k^i := \int_{a_k^i}^{b_k^i} p_k(x_k) dx_k. \tag{42}$$

In general, this has to be evaluated numerically. For the special case of uniform distribution on a rectangular $\mathbf{X}$, we obtain:

$$P_k^i = \frac{b_k^i - a_k^i}{b_k - a_k}.$$

We define the *importance* I_k^i of the dimension k of the element $\mathbf{X}^i$ to be:

$$I_k^i = P_k^i / \ell_k^i. \tag{43}$$

Intuitively, the importance of a particular dimension is inversely proportional to the inferred length scale and proportional to the probability mass along that dimension trapped within the stochastic element. Thus, if $\mathbf{X}^i$ needs refinement (i.e. satisfies Eq. (40)), we cut it perpendicular to the most important dimension k^*, given by:

$$k^* = \arg\max_k I_k^i. \tag{44}$$

In order to have two new elements with the same probabilities of occurrence, the splitting point is given by the median of the marginal conditional

distribution of $\mathbf{X}^i$ along dimension k, $p_k^i(x_k)$ defined by:

$$p_k^i(x_k) = \frac{p_k(x_k)}{\int_{a_k^i}^{b_k^i} p_k(x'_k)dx'_k} 1_{[a_k^i, b_k^i]}(x_k). \tag{45}$$

This is a root finding problem that can easily be solved using a bisection algorithm. For the special case of the uniform distribution, the splitting point trivially becomes:

$$x_k^* = \frac{1}{2}(a_k^i + b_k^i).$$

Remark 2. The particular splitting criterion based on the inferred length scales is not the only possibility. Despite being intuitively appealing, it remains an ad hoc choice. Nevertheless, its computational evaluation time is negligible and we have empirically shown that it results in decompositions that concentrate around important features of the response. Of course, its performance depends crucially on predicting correctly the length scales.

2.5. Collection of the observations

In this section, we discuss how the data within an element are collected. We have to consider two distinct cases:

1. No data have been observed yet and we only have a single element (i.e. $\mathbf{X}$ itself).
2. We have obtained a fit of the response over an element $\mathbf{X}^i$ based on N^i observations
$$\mathcal{D}^i = \{(\mathbf{x}^{i,(n)}, \mathbf{y}^{i,(n)})\}_{n=1}^{N^i},$$
and we have decided to split it in two elements $\mathbf{X}^{i,1}$ and $\mathbf{X}^{i,2}$ so that
$$\mathbf{X}^i = \mathbf{X}^{i,1} \cup \mathbf{X}^{i,2} \text{ and } \mathbf{X}^{i,1} \cap \mathbf{X}^{i,2} = \emptyset.$$

Let $N \geq 1$ be the *maximum number of observations per element* we wish to consider within each element and $\delta > 0$ be the desired uncertainty tolerance of each element (see Eq. (40)). We deal with the first case (no observations made so far), by simply observing N random data points drawn from the input probability distribution $p(\mathbf{x})$. In the second case, we wish to utilize the MGP we already have for $\mathbf{X}^i$, in order to make the most informative selection of new data points. This procedure is known in the literature as *Experimental Design* (ED).

The ED problem can be formulated in a Bayesian framework in terms of maximizing the expectation of a utility function (see [35] for a good review of Bayesian ED). If we observe the data points one by one and update the model each time, then the procedure is termed *Sequential Experimental Design* (SED). In the machine learning literature SED is known as *Active Learning* (AL). According to MacKay [29], if the utility function we choose is the change in entropy of the posterior of the hyper-parameters $\boldsymbol{\theta}$, then - under the evidence approximation - the most informative input point corresponds to the one that maximizes the predictive variance of the model. This criterion is termed *Active Learning MacKay* (ALM). An alternative to ALM is Cohn's criterion (ALC) [36], which proceeds by choosing the input point that maximizes the expected change in output predictive variance over the whole domain. ALC has the advantage that it allows one to weight the input space by a probability distribution, which in our setting would naturally be the input probability distribution of the element $\mathbf{X}^i$. ALC has also been numerically shown to perform better than ALM (for a comparison of ALM and ALC see [37] and the corresponding discussion in [38]). However, ALC is not based on a decision theoretic foundation and it is much harder to implement. In this work - mainly for computational purposes - we choose to work with ALM. We now, describe its extension to the multi-output case.

We start, by splitting the observed data in two sets $\mathcal{D}^{i,l},\ l = 1, 2$ according to which element the inputs belong to, i.e.

$$\mathcal{D}^{i,l} = \{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}^i : \mathbf{x} \in \mathbf{X}^{i,l}\},\ l = 1, 2.$$

Let $\boldsymbol{\theta}^*$ be the hyper-parameters of the MGP over $\mathbf{X}^i$ and $\sigma^2_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*)$ be the corresponding predictive variance of the r-th output at $\mathbf{x} \in \mathbf{X}^i$ given by Eq. (15). Throughout the SED procedure, the hyper-parameters will be kept constant. Without loss of generality, we work with the left child of $\mathbf{X}^i$, $\mathbf{X}^{i,1}$. The right child is treated similarly. We will be sequentially observing $\mathbf{x}^{\text{new},m}$ and the corresponding responses $\mathbf{y}^{\text{new},m} = \mathbf{f}(\mathbf{x}^{\text{new},m})$ for $m = 1, 2, \ldots$. Let the set of observations residing in $\mathbf{X}^{i,1}$ be:

$$\mathcal{D}^{i,1,n} = \mathcal{D}^{i,1} \cup \{\mathbf{x}^{\text{new},m} : m = 1, \ldots, n\}, n \geq 1,$$

where $\mathcal{D}^{i,1,0} = \mathcal{D}^{i,1}$. Denote by $\sigma^2_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*, \mathcal{D}^{i,1,n})$ the predictive variance of the r-th output when $\mathcal{D}^{i,1,n}$ is taken into account. From Eq. (15), it is apparent that $\sigma^2_{f_r}(\mathbf{x}; \boldsymbol{\theta}^*, \mathcal{D}^{i,1,n})$ depends only on the observed input points and not on the responses. Furthermore, since $\boldsymbol{\theta}^*$ remains constant, the inverse covariance

matrix can be estimated sequentially at each step without the need to perform a Cholesky decomposition (see [21]). The extension of ALM to the multi-output case is as follows: given $\mathcal{D}_{i,1}^{n}$, observe the input point $\mathbf{x}^{\text{new},n+1} \in \mathbf{X}^{i,1}$ that maximizes the joint uncertainty of all outputs:

$$\sigma_{\mathbf{f}}^{2}(\mathbf{x};\boldsymbol{\theta}^{*};\mathcal{D}^{i,1,n}) = \frac{1}{M}\sum_{r=1}^{M}\sigma_{f_r}^{2}(\mathbf{x};\boldsymbol{\theta}^{*},\mathcal{D}^{i,1,n}). \tag{46}$$

That is,

$$\mathbf{x}^{\text{new},n+1} = \arg\max_{\mathbf{x}\in\mathbf{X}^{i,1}} \sigma_{\mathbf{f}}^{2}(\mathbf{x};\boldsymbol{\theta}^{*};\mathcal{D}^{i,1,n}). \tag{47}$$

In an effort to introduce a bias from the input probability distribution, we suggest using:

$$\mathbf{x}^{\text{new},n+1} = \arg\max_{\mathbf{x}\in\mathbf{X}^{i,1}} \sigma_{\mathbf{f}}^{2}(\mathbf{x};\boldsymbol{\theta}^{*};\mathcal{D}^{i,1,n})p(\mathbf{x}), \tag{48}$$

which causes low probability regions to be ignored. Of course, for the uniform case the two criteria are equivalent. We stop, either if N data points have been collected in $\mathcal{D}^{i,1,n}$, or if:

$$\sigma_{\mathbf{f},p^{i,1}}^{2}(\mathcal{D}^{i,1,n})P(\mathbf{X}^{i,1}) \leq \delta, \tag{49}$$

where $\sigma_{\mathbf{f},p^{i,1}}^{2}(\mathcal{D}^{i,1,n})P(\mathbf{X}^{i,1})$ is the expectation of $\sigma_{\mathbf{f}}^{2}(\mathbf{x};\boldsymbol{\theta}^{*};\mathcal{D}^{i,1,n})$ with respect to the conditional probability $p^{i,1}(\mathbf{x})$ of $\mathbf{X}^{i,1}$ (in the same spirit as it was used in Section 2.4).

The optimization problem in Eq. (48) is relatively hard and involves several local maxima. Instead of solving it with a direct method, we use a simple Monte Carlo procedure to obtain an approximate solution. We draw $N_{textALM}$ random samples in $\mathbf{X}^{i,1}$, evaluate the product of the predictive variances and the input probability density (Eq. (46)) and select the one yielding the greatest result. This is affordable, since $\sigma_{\mathbf{f}}^{2}(\mathbf{x};\boldsymbol{\theta}^{*};\mathcal{D}^{i,1,n})$ is cheap to evaluate.

2.6. A complete view at the framework

In this final section, we put together the building blocks of our scheme and discuss the algorithmic details and possible parallelization strategies. The basic input required is the maximum number of observations per element N and the tolerance $\delta > 0$, used for the refinement criterion (Eq. (40)) as

Algorithm 1 The complete surrogate building framework

$\mathcal{U} \leftarrow \{(\mathbf{X}, \emptyset, \emptyset)\}$.
$\mathcal{C} \leftarrow \emptyset$.
while $\mathcal{U} \neq \emptyset$ **do**
  Remove $(\mathbf{X}^i, \mathcal{D}^i, \mathcal{M}^i)$ from $\mathcal{U}$.
  if $\mathcal{M}^i = \emptyset$ **then**
    Observe N random points drawn from $p^i(\mathbf{x})$ Eq. (18).
  else
    while $|\mathcal{D}^i| < N$ or Eq. (49) not satisfied for δ **do**
      Add an observation to $\mathcal{D}^i$ using the ALM procedure (Eq. (48)).
      Update $\mathcal{M}^i$ to take into account the new data in $\mathcal{D}^i$.
    end while
  end if
  Refit the hyper-parameters of $\mathcal{M}^i$ using only the data in $\mathcal{D}^i$ (Section 2.1).

  if Refinement criterion of Eq. (40) is satisfied for δ **then**
    Split $\mathbf{X}^i$ in $\mathbf{X}^{i,1}$ and $\mathbf{X}^{i,2}$ according to Eq. (44).
    Let $\mathcal{D}^{i,1}$ and $\mathcal{D}^{i,2}$ to be the set observations residing in $\mathbf{X}^{i,1}$ and $\mathbf{X}^{i,2}$, respectively.
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{(\mathbf{X}^{i,1}, \mathcal{D}^{i,1}, \mathcal{M}^i), (\mathbf{X}^{i,2}, \mathcal{D}^{i,2}, \mathcal{M}^i)\}$.
  else
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathbf{X}^i, \mathcal{D}^i, \mathcal{M}^i)\}$.
  end if
end while

well as the stopping criterion of ALM (Eq. (49)). An additional input is the number of MC samples used to approximate the solution to Eq. (48) (last paragraph of Section 2.5), which we fix to $N_{\text{ALM}} = 10000$.

Our scheme works in one element cycles that comprise of collecting observations (randomly or using ALM (Section 2.5)), fitting (Section 2.1) and adapting (Section 2.4). Let us denote with $\mathbf{X}^i$ a stochastic element, $\mathcal{D}^i$ the observations made on $\mathbf{X}^i$ and $\mathcal{M}^i$ the MGP fitted over $\mathbf{X}^i$ using $\mathcal{D}^i$. Let $\mathcal{C}$ be the set of triplets $(\mathbf{X}^i, \mathcal{D}^i, \mathcal{M}^i)$ for which the refinement criterion Eq. (40) is not satisfied. We will refer to $\mathcal{C}$ as the set of *completed triplets*. The rest of the triplets are put in $\mathcal{U}$, called the set of *uncompleted triplets*. With $|\mathcal{D}^i|$ we denote the number of observations inside $\mathcal{D}^i$. Algorithm 1 provides a serial implementation of the scheme.

Parallelization of Algorithm 1 is relatively easy. Each node p, has its own set of completed $\mathcal{C}_p$ and uncompleted $\mathcal{U}_p$ elements. Initially the root node $p = 0$ starts as in Algorithm 1 and the rest with $\mathcal{U}_p = \emptyset, \mathcal{C}_p = \emptyset, p \neq 0$. Then, everything proceeds as in Algorithm 1 with load re-balancing at the end of each outer iteration (uncompleted elements are sent to processors with $\mathcal{U}_p = \emptyset$).

Remark 3. The choice of the maximum number of samples per element N is a crucial parameter to the scheme. Its optimal value depends in a complicated way on the (a priori unknown) smoothness of the underlying response as well as the number of hyper-parameters S. Its importance is more evident on the very first element of the scheme because it drives the rest of the tree construction as well as the Active Learning procedure. If a small value is used, then local features may be lost, while a very big value may result in redundant information. Similar problems are present in practically all UQ schemes. For example, ME-gPC depends on the polynomial degree and ASGC depends on which level of the Sparse Grid is adaptivity initiated. On the other hand, N makes our method computationally tractable, since it bounds above the dimensions of the covariance matrices that need to be inverted. A theoretical analysis of the optimal value of N is highly desirable, but clearly beyond the scope of the present work. In the engineering problems that we are interested in, one usually already has a rough idea about the smoothness of the problem based on some preliminary simulations. For smooth problems, using $N \approx 2K$, where K is the number of input dimensions gives satisfying results (see the Elliptic and Natural Convection numerical examples in Section 3). For problems with local features, a slightly bigger value must be used. Empirically, we fix δ to a high value (e.g. $\delta \approx 10^{-1}$), we start with $N = 2K$ and increase N gradually until the results do not change any more. For this final N, we decrease δ and resume the scheme.

3. Numerical Examples

All examples are run on massively parallel computers at the National Energy Research Scientific Computing Center (NERSCC). The parallelization strategy is straightforward: each processor is assigned to work with a single element. The communication burden between the processes is minimal. Our implementation utilizes extensively the Trilinos library [39] as well as GSL [40].

The ultimate goal of the numerical examples is to demonstrate that the method can:

1. learn non-stationary surfaces,
2. deal with discontinuities,
3. identify localized features of the response and
4. reduce sampling frequency on unimportant input dimensions.

Whenever possible, we will compare our results with Sparse Grid Collocation (SGC) and Adaptive Sparse Grid Collocation (ASGC) [13]. Each method will be evaluated by considering an error measure of the predictive surface or of the statistics, as a function of the number of sample points used. In Section 3.1, we investigate the performance of our method in learning three synthetic functions. In Sections 3.2, 3.3, 3.4, we apply our method to UQ problems. In all problems, the underlying input probability distribution $p(\mathbf{x})$ is understood to be the uniform distribution over the input domain. The covariance function we use, is the SE with a nugget $g^2 = 10^{-6}$ (The nugget is required for numerical stability. See the discussion in Appendix A for more details.). All tasks start with a single element (the input domain itself) and N random samples drawn from the input distribution. N, is also the maximum number of samples taken within an element and is different for each example (See Remark 3 for to see how N can be chosen). From that point, the algorithm proceeds until a pre-specified tolerance $\delta > 0$ is reached. The refinement criterion is given by Eq. (40) for $q = 1$. The same tolerance δ is used to stop the ALM procedure of Section 2.5 (see Eq. (49)). The solution to the optimization problem of ALM (Eq. (48)) is approximated by drawing $N_{\text{ALM}} = 10000$ samples in $\mathbf{X}^i$, evaluating $\sigma^2_{\mathbf{f}}(\mathbf{x}; \boldsymbol{\theta}^*; \mathcal{D}^{i,1,n})$ and selecting the one with the maximum value. The parameters of the method are g, N, N_{ALM} and δ.

3.1. Simple Validation Examples

The purpose of this section is to demonstrate using simple, single-output functions, the claims $1 - 4$ made at the beginning of this section. The three synthetic functions we are going to use have been introduced in [38]. The performance of each run is evaluated by comparing the predictive mean $\mu_{f_r}(\mathbf{x})$ to the true response. The error measure of choice here is the Mean Square Error (MSE) of $S = 10^5$ random samples drawn from $p(\mathbf{x})$. Specifically, MSE

is defined to be

$$\mathrm{MSE}\left(\mu_{f_r}(\cdot)\right) := \frac{1}{SM}\sum_{s=1}^{S}\sum_{r=1}^{M}\left(\mu_{f_r}(\mathbf{x}^{(s)}) - f_r(\mathbf{x}^{(s)})\right)^2, \tag{50}$$

where $\mathbf{x}^{(s)}, s = 1, \ldots, S$ are random samples from $p(\mathbf{x})$. Those samples were not used in the fitting procedure, hence MSE is a measure of the predictive capabilities of the regression method.

1D non-stationary, discontinuous function. Consider the real function:

$$f_1(x) = \begin{cases} \sin\left(\frac{\pi x}{5}\right) + \frac{1}{5}\cos\left(\frac{4\pi x}{5}\right), & x \leq 10 \\ \frac{x}{10} - 1, \text{otherwise} \end{cases}, \tag{51}$$

on the domain $\mathbf{X} = [0, 20]$. For $x \leq 10$, it varies with two different frequencies. For $x > 10$ it is linear and finally it has a discontinuity at $x = 10$.

We learn this function with our framework using $N = 5$ until various tolerances are reached. Fig. 1 compares the MSE of MGP with ASGC for various numbers of observations. The observations shown for MGP correspond to tolerances of $\delta = 10^{-1}, 10^{-2}, 10^{-5}, 10^{-6}, 10^{-7}$ and 10^{-8}. The ϵ parameter of ASGC (see [13]) is a lower bound of the sparse grid surpluses. The bigger ϵ is, the more samples ASGC skips. As ϵ goes to zero, the ASGC approaches SGC. For large values of ϵ though, ASGC fails to converge. Hence, ϵ determines the balance between *exploration* and *exploitation* is ASGC. It is apparent that ASGC is out-performed by MGP by almost an order of magnitude. Fig. 2 plots the predictive mean $\mu_{f_1}(x)$ with 95% error bars for $\delta = 10^{-2}, 10^{-4}$ and 10^{-6} along with the true response $f_1(x)$ where the symbols mark the position of the observed data (left column). Notice, that the linear part is already captured at $\delta = 10^{-2}$ (13 observations) and that the region $x > 10$ is not sampled any further. Another important observation is that the error bars are maximized in regions of space where the true error is bigger. This fact is the empirical justification of their usage in the SED framework of Section 2.5. As the lower levels of tolerance are reached, more and more samples are collected inside the important regions and the discontinuity is finally resolved. The right column of the same figure, plots the value of the inferred length scale ℓ as a function of x (one length scale at each element). The linear region is treated as a single element with a large length scale, while the rest of the domain is fragmented in smaller elements with small length scales.

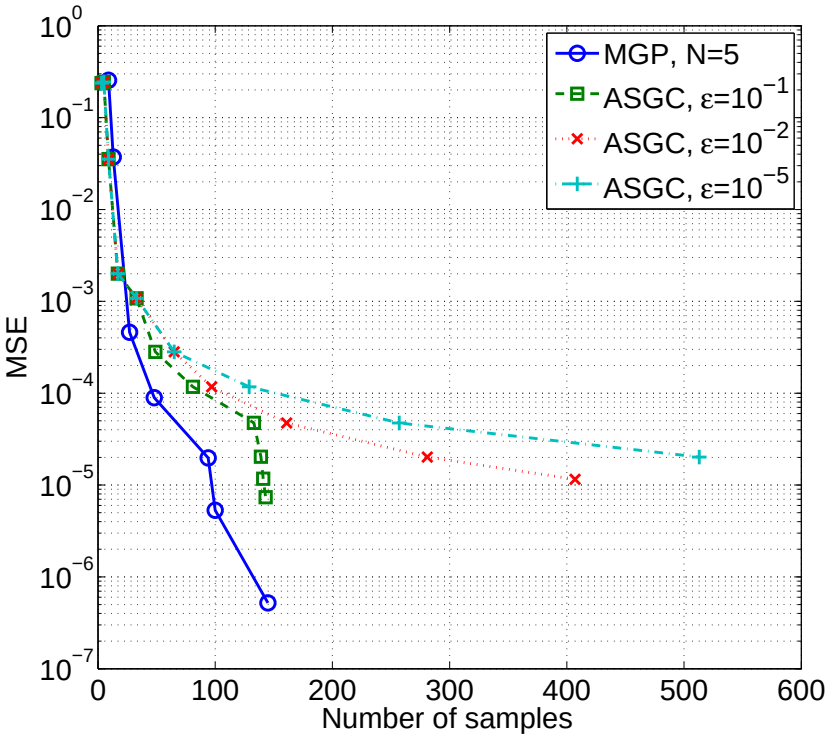


Figure 1: The MSE in the prediction of $f_1(x)$ as a function of the observed samples for MGP and ASGC for various ϵ.

2D function with local features. Let us now consider a two-dimensional real function:

$$f_2(x_1, x_2) = x_1 \exp\left\{-x_1^2 - x_2^2\right\}, \tag{52}$$

on $\mathbf{X} = [-2, 6]^2$. This function is peculiar, in the sense that it has two localized features inside the box $[-2, 2]^2$, while it is practically zero everywhere else. The choice of N in this example plays an important role since it determines the starting point of our algorithm. We have numerically verified that for $N = 5$ there is a high probability of not observing the localized features. For $N = 10$ and 20 the features are observed, albeit after a few fluctuations which result in a higher number of observations being made. ASGC starting from Level 1 (any ϵ) fails to correctly identify the location of the localized features, since it does not sample inside $[-2, 2]^2$. On the other hand, SGC requires a very large number of observations. Here, we choose to report our

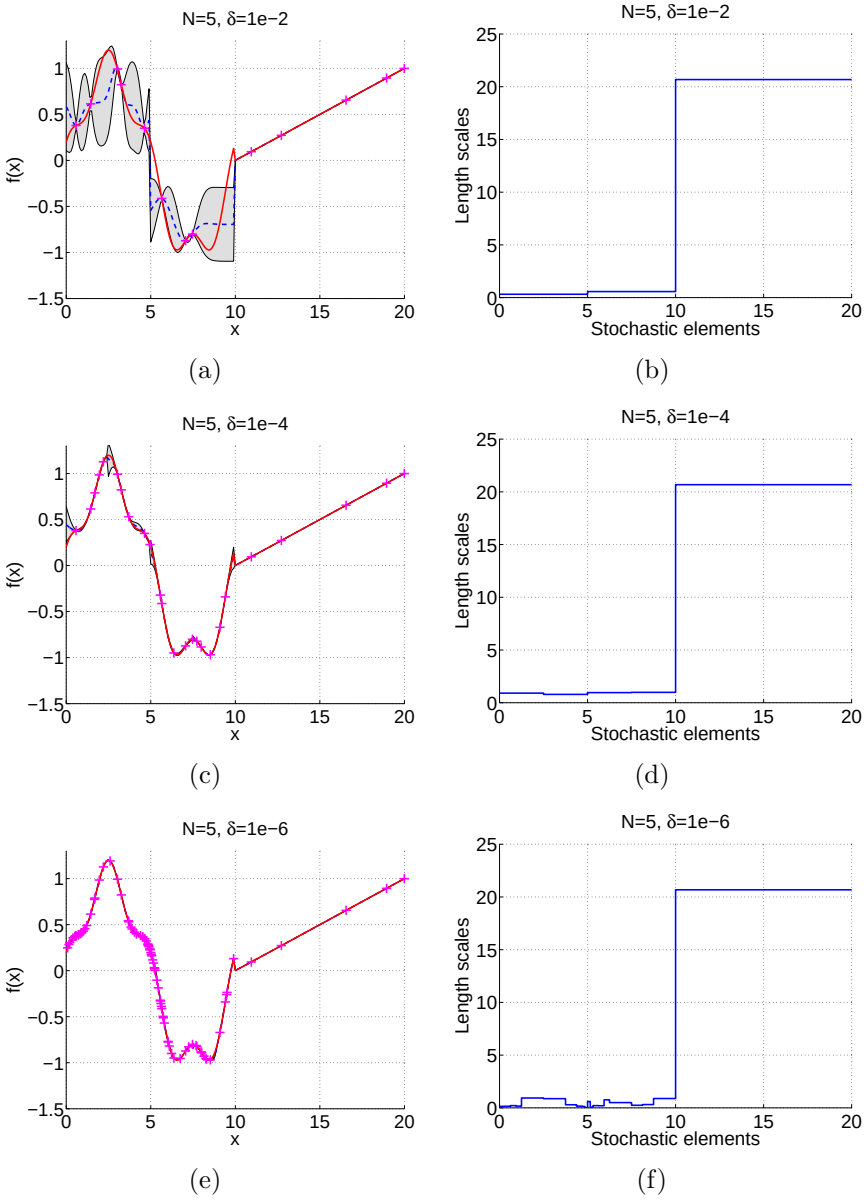


Figure 2: Left column (a, c, e): comparison of the predictive mean $\mu_{f_1}(x)$ (dashed blue line) and 95% error bars (shaded grey area) with true response $f_1(x)$ (solid red line), where the symbols mark the observed data. Right column (b, d, f): predicted length scale across the domain. The rows correspond to tolerances $\delta = 10^{-2}, 10^{-4}$ and 10^{-6} with number of samples gathered 13, 25 and 94, respectively.

results for $N = 50$. Fig. 3 plots the MSE for MGP and SGC as a function of the number of observations. The MSE of ASGC is not reported since it fails to identify the localized features when it starts from Level 1. The observations shown for MGP correspond to tolerances of $\delta = 10^{-3}, 10^{-4}, 10^{-5}$ and 10^{-6}. As expected, SGC is out-performed by more than two orders of magnitude. Fig. 4 shows the contour of the predictive mean $\mu_{f_2}(x_1, x_2)$ for tolerances $\delta = 10^{-4}, 10^{-5}$ and 10^{-6} (right column) along with the decomposition of the stochastic domain. The left column depicts the corresponding observed input points (left column). Notice how the density of the observations is increasing in the important regions as lower δ's are reached, gradually revealing the local features.

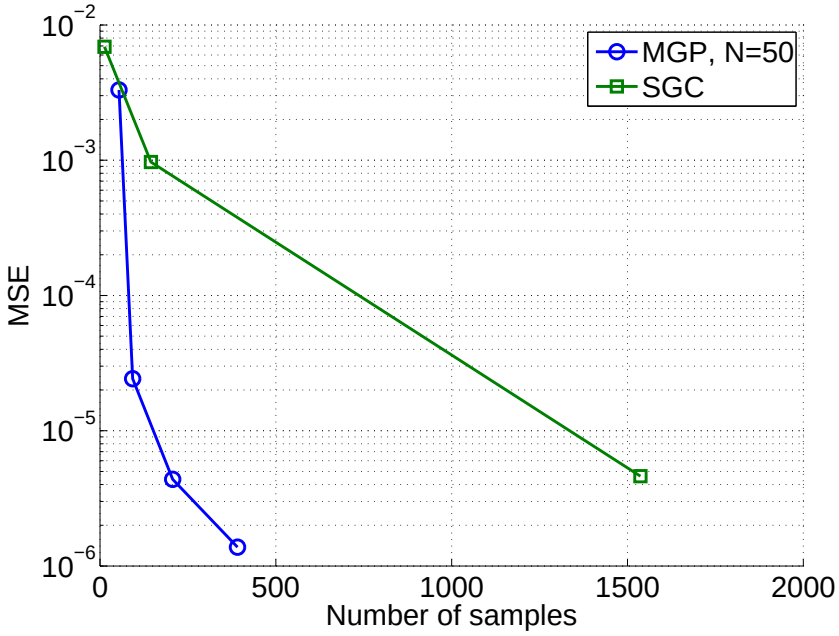


Figure 3: The MSE in the prediction of $f_2(\mathbf{x})$ as a function of the observed samples for MGP and SGC. ASGC ($\epsilon = 10^{-3}$) is not reported since it fails to identify the localized features.

6D function with unimportant dimensions. Finally, we consider the six-dimensional real function:

$$f_3(x_1, x_2, x_3, x_4, x_5, x_6) = \exp\left\{\sin\left((0.9(x_1 + 0.48)^{10}\right)\right\} + x_2 x_3 + x_4, \quad (53)$$

on the hypercube $\mathbf{X} = [0, 1]^6$. f_3 varies wildly as a function of x_1 (see (a) of Fig. 5), it is linear in x_4, quadratic with respect to x_2 and x_3 and constant for x_5 and x_6. We learn it using our scheme with $N = 10$. Fig. 6 plots the MSE for MGP, SGC and ASGC as a function of the number of observations. ASGC is out-performed by at least an order of magnitude. In Fig. 5, we analyze the distribution of the observed input points. In particular, we plot the histogram of the projection of the observed inputs on x_1 (b), x_5 (c) and x_6 (d) axes. Notice that MGP increases the sampling density in important regions with respect to x_1 while x_5 and x_6 are sampled uniformly. The histograms for x_2, x_3 and x_4 are similar to x_5 and x_6.

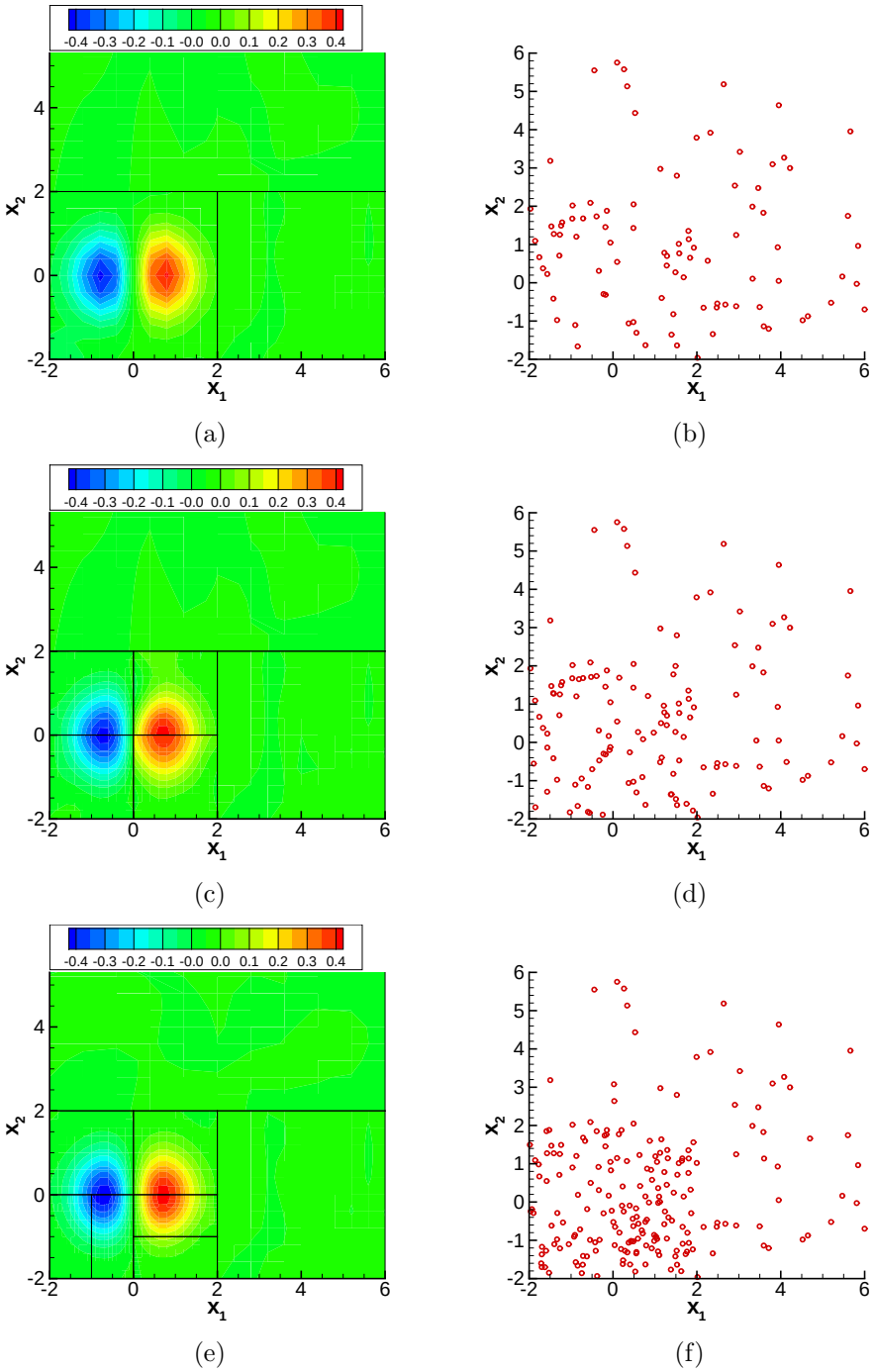


Figure 4: Left column ($a,\ c,\ e$): predictive mean for $f_2(\mathbf{x})$ of the MGP ($N = 50$) and decomposition of the domain for $\delta = 10^{-4}, 10^{-5}$ and 10^{-6} (top to bottom). Right column ($b,\ d,\ f$): observations made for the same δ.

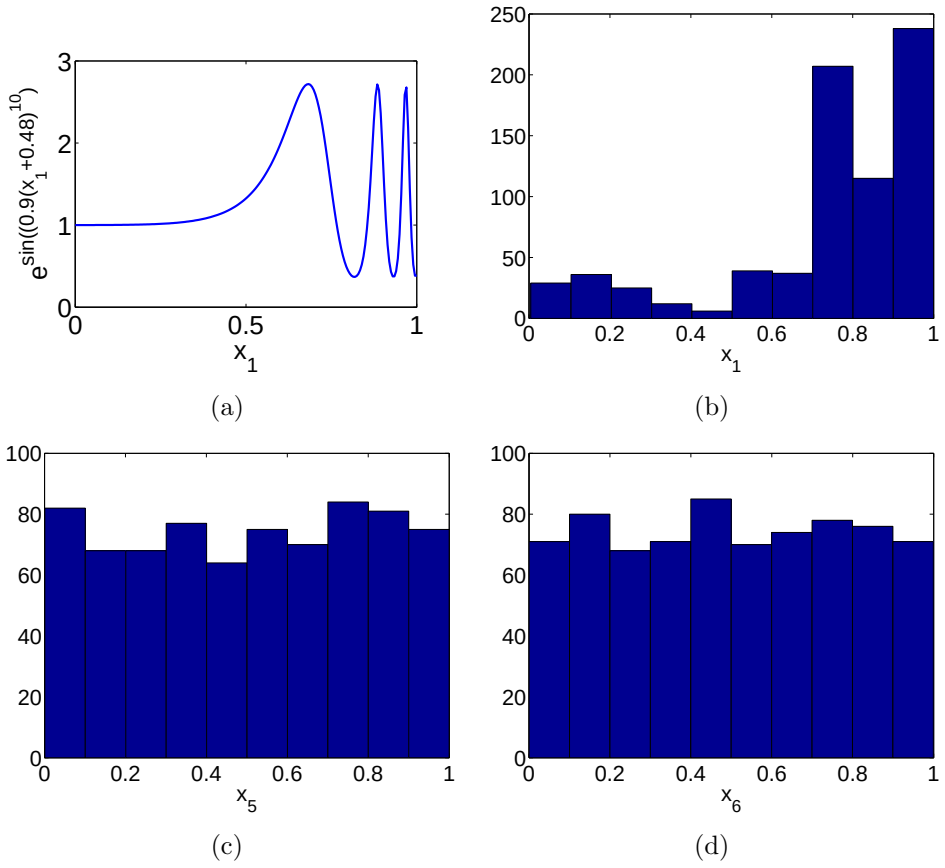


Figure 5: (a) The additive part of $f_3(x_1, \ldots, x_6)$ that depends on x_1; (b), (c) and (d) are histograms of the projections of the observed inputs on the x_1, x_5 and x_6 axes, respectively.

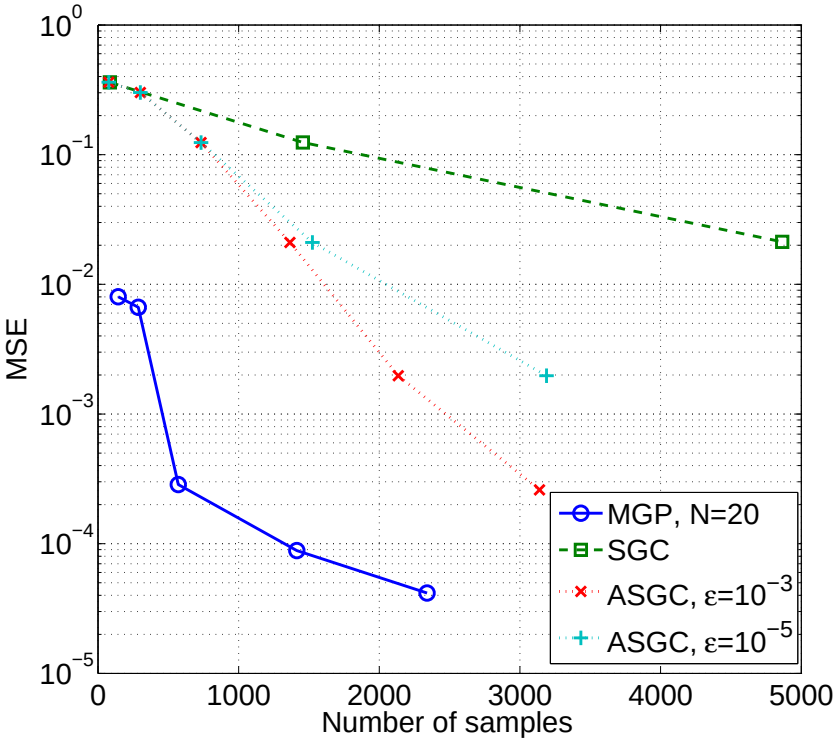


Figure 6: The MSE in the prediction of $f_3(\mathbf{x})$ as a function of the observed samples for MGP, SGC and ASGC ($\epsilon = 10^{-3}$).

3.2. Krainchnan-Orszag three-mode problem

Consider the system of ordinary differential equations [6]:

$$\begin{aligned} \frac{dy_1}{dt} &= y_1 y_3, \\ \frac{dy_2}{dt} &= -y_2 y_3, \\ \frac{dy_3}{dt} &= -y_1^2 + y_2^2, \end{aligned}$$

subject to random initial conditions at $t = 0$. This dynamical system is particularly interesting because the response has a discontinuity at the planes $y_1(0) = 0,\ y_2(0) = 0$. The deterministic solver we use is a 4th order Runge-Kutta method as implemented in GNU Scientific Library [40]. We solve the system for the time interval $[0, 10]$ and record the response at time step intervals of $\Delta t = 0.01$. This results in a total of $M = 300$ outputs (100 for each of the three dimensions of the response). We will consider three different cases of increasing difficulty with one, two and three input dimensions. The results we obtain will be compared to a MC estimate with 10^6 samples. Let the MC mean and variance be $m_{r,\mathrm{MC}}$ and $v_{r,\mathrm{MC}}$, respectively, $r = 1, \ldots, 300$. The error of the statistics will be evaluated using the (normalized) L_2 *norm of the error in variance* defined by:

$$\mathrm{E_{L2}} = \frac{1}{M} \sum_{r=1}^{M} \left(v_{r,\mathrm{MC}} - \mu_{v_r} \right)^2, \tag{54}$$

where μ_{v_r} is the predictive mean of v_r (Eq. (32)). The results are compared with SGC and ASGC.

One-dimensional Problem. In the one-dimensional case, we define the stochastic initial conditions by:

$$y_1(0) = 1,\ y_2(0) = 0.1x,\ y_3(0) = 0,$$

where

$$x \sim U([-1, 1]),$$

with $U([-1, 1])$ being the uniform probability distribution over $[-1, 1]$. This stochastic problem has a discontinuity at $x = 0$. We solve it for $N = 5$. Fig. 7 shows the L_2 norm of the error in variance for MGP, SGC and ASGC

as a function of the number of observations. ASGC for $\epsilon = 10^{-1}$ fails to converge and so it is not reported. MGP slightly out-performs ASGC and SGC, especially when just a few samples are used. Fig. 8 depicts the prediction of y_2 $(t = 10)$ and y_3 $(t = 10)$ at levels of tolerance $\delta = 10^{-3}, 10^{-5}$ and 10^{-7}. Again, we observe that the error bars are qualitatively equivalent to the true error. Notice how the discontinuity is gradually resolved. Fig. 9 plots the predictive mean and variance of $y_3(t)$ as a function of time t along with 95% error bars (see Eqs. (28) and (31)) and compares them with the MC predictions. The error bars of the statistics are qualitatively correct but - as expected by the independence assumption (Section 2.2) - they are over-estimated. This situation is more pronounced in the predictions for the statistics of the two and three dimensional problems.

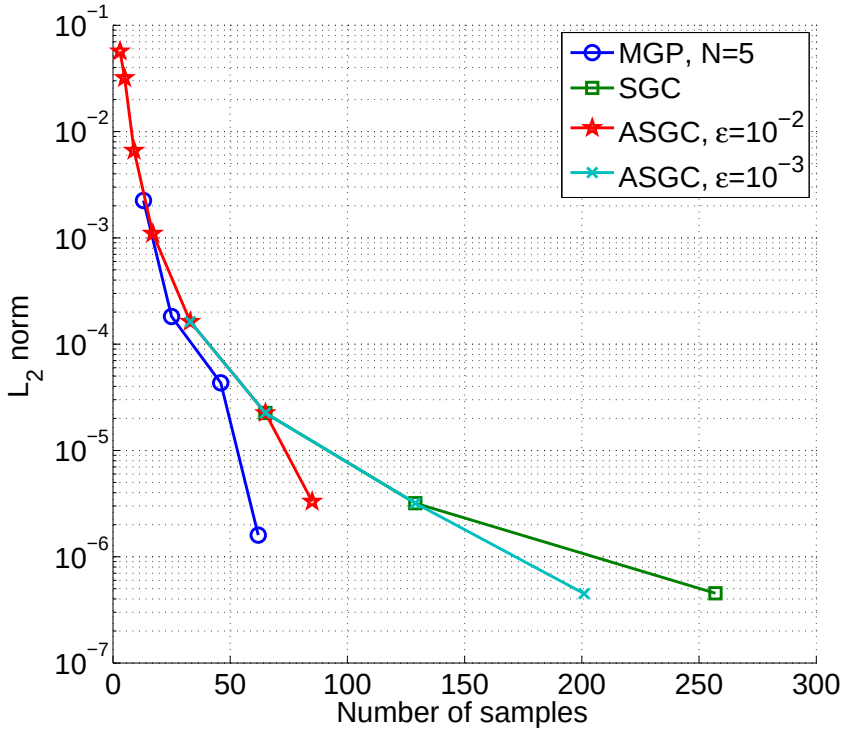


Figure 7: KO-1: the L_2 norm of the error in variance as a function of the observed samples for MGP, SGC and ASGC.

Two-dimensional Problem. For the two-dimensional problem, the stochastic initial conditions are defined by:

$$y_1(0) = 1,\ y_2(0) = 0.1x_1,\ y_3(0) = x_2,$$

where

$$x_i \sim U([-1, 1]),\ i = 1, 2.$$

This problem has a line discontinuity at $x_1 = 0$. We run the MGP framework for $N = 10$. Fig. 10 shows the L_2 norm of the error in variance for MGP, SGC and ASGC as a function of the number of observations. At this example, the performance of MGP and ASGC ($\epsilon = 10^{-2}$) is approximately the same. Fig. 11 depicts the prediction at y_3 ($t = 10$) along with the stochastic elements at levels of tolerance $\delta = 10^{-3},\ 10^{-5}$ and 10^{-7}. As a lower tolerance is reached, the stochastic mesh adapts around the discontinuity increasing the sampling density. Fig. 12 plots the predictive mean and variance of $y_3(t)$ as a function of time t along with 95% error bars and compares it with the MC prediction. Again, we notice that the error bars are over-estimated. Finally, by using 10^4 samples of the surrogate, we provide a kernel density approximation to the probability density function (PDF) of y_2 ($t = 10$) and y_3 ($t = 10$) and compare it to an MC estimate with the same number of samples (Fig. 13).

Three-dimensional Problem. The three-dimensional problem is defined to have initial conditions:

$$y_1(0) = x_1,\ y_2(0) = x_2,\ y_3(0) = x_3,$$

where

$$x_i \sim U([-1, 1]), i = 1, 2, 3.$$

We run our framework for $N = 20$. Fig. 14 shows the L_2 norm of the error in variance for MGP, SGC and ASGC as a function of the number of observations. ASGC with $\epsilon = 10^{-1}$ fails to converge. MGP out-performs ASGC. Fig. 15 plots the predictive mean and variance of $y_3(t)$ as a function of time t along with 95% error bars and compares it with the MC prediction. Finally, Fig. 16 plots the kernel density estimate of the PDF of y_2 ($t = 10$) and y_3 ($t = 10$) using 10^4 samples of the surrogate.

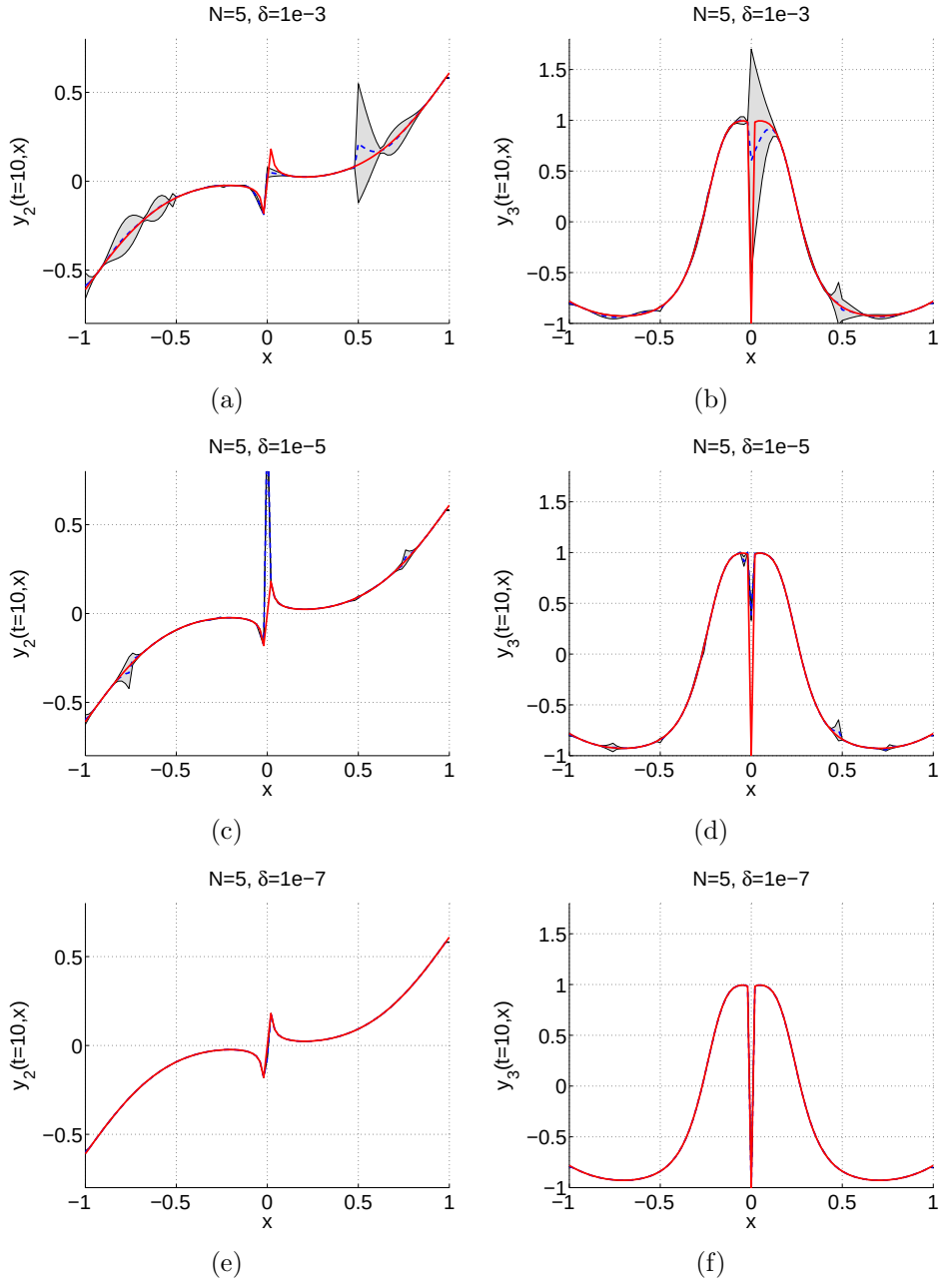


Figure 8: KO-1: prediction (dashed blue) with 95% error bounds for tolerances (top to bottom) $\delta = 10^{-3}, 10^{-5}$ and 10^{-7} versus the true response (solid red) for y_2 $(t = 10)$ (left column, $a,\ c,\ e$) and y_3 $(t = 10)$ (right column, $b,\ d,\ f$).

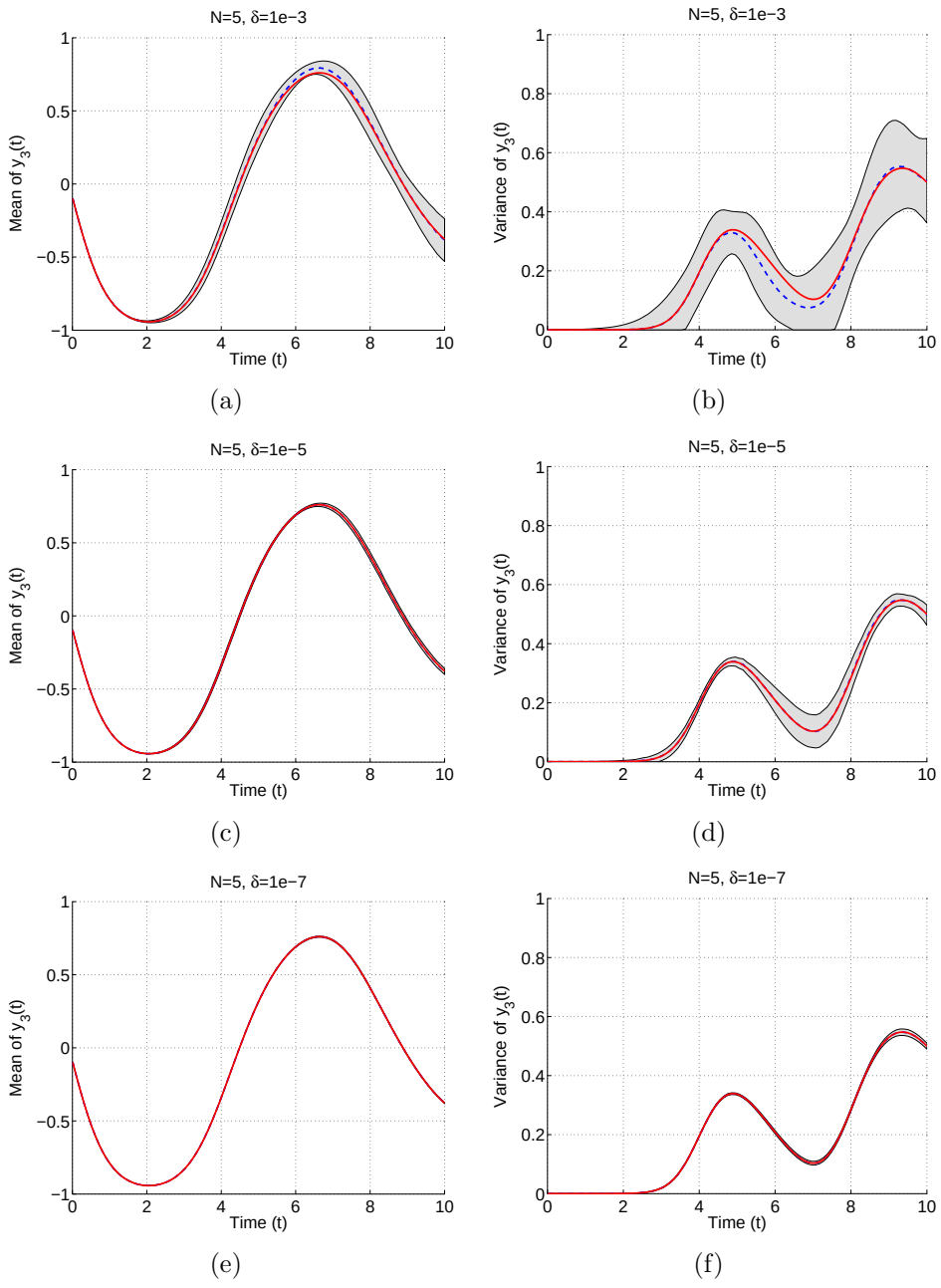


Figure 9: KO-1: predictive mean (dashed blue) versus MC estimate (solid red) of the mean (left column, $a,\ c,\ e$) and variance (right column, $b,\ d,\ f$) of $y_3(t)$ with 95% error bounds for tolerances (top to bottom) $\delta = 10^{-3},\ 10^{-5}$ and 10^{-7}.

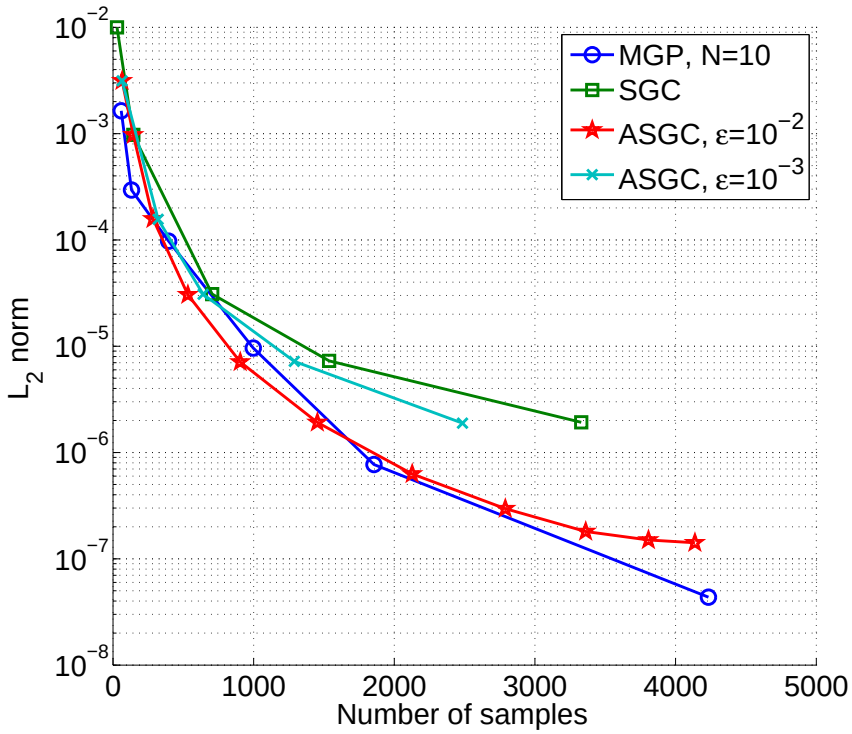


Figure 10: KO-2: the L_2 norm of the error in variance as a function of the observed samples for MGP, SGC and ASGC.

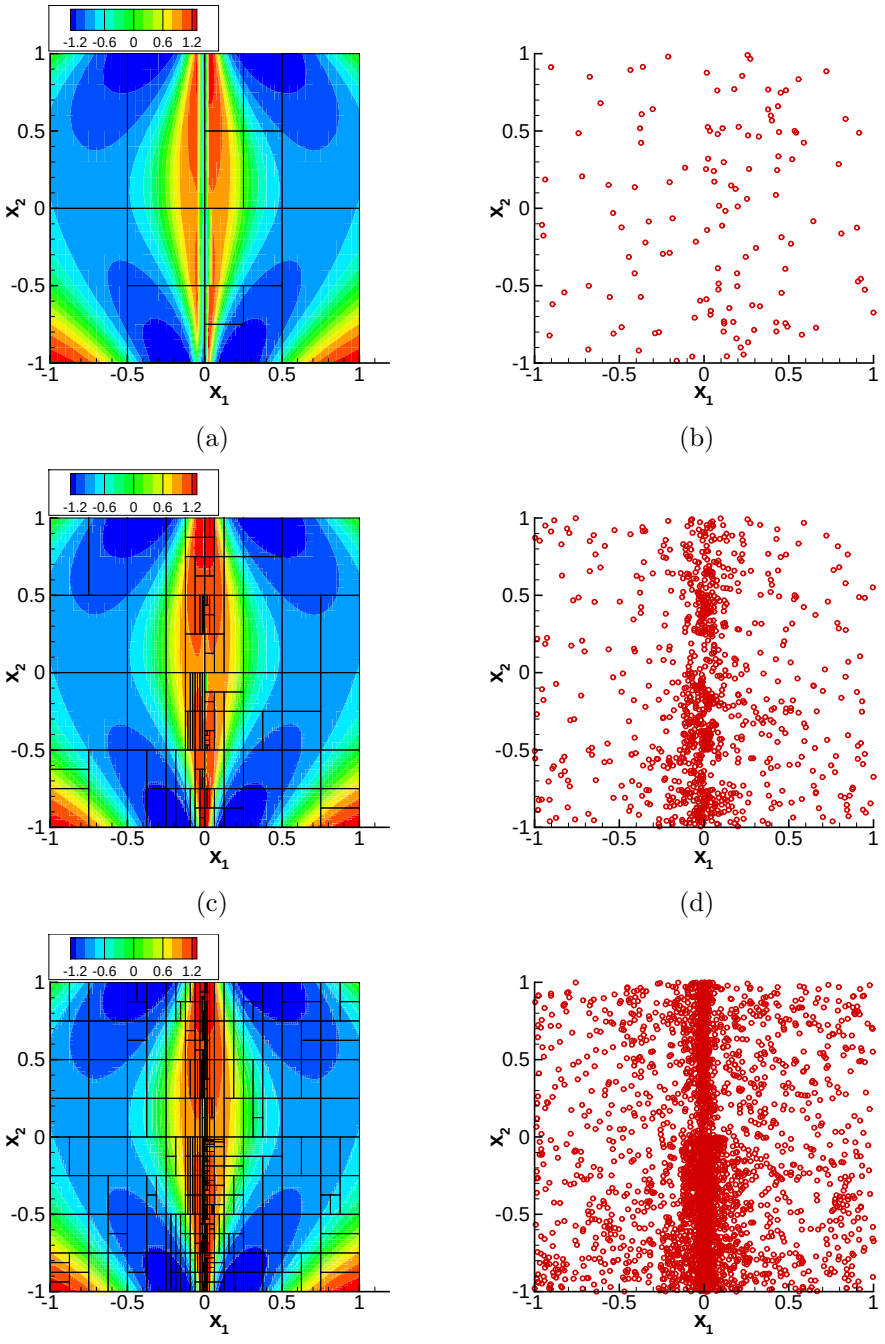


Figure 11: KO-2: The prediction at y_3 ($t = 10$) with the stochastic elements (left column, $a,\ c,\ e$) and the observed samples (right column, $b,\ d,\ f$) for tolerances (top to bottom) $\delta = 10^{-3},\ 10^{-5}$ and 10^{-7}.

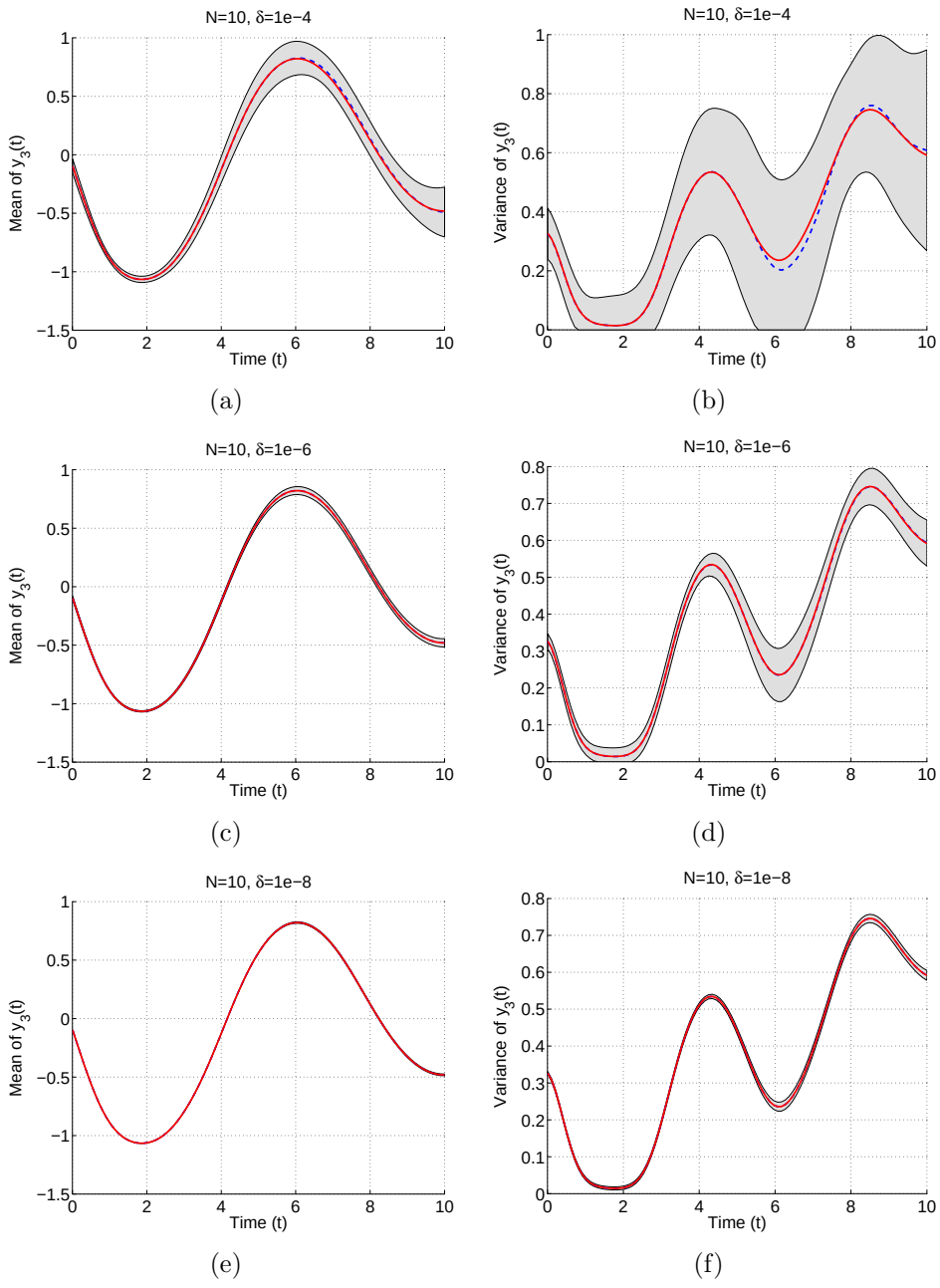
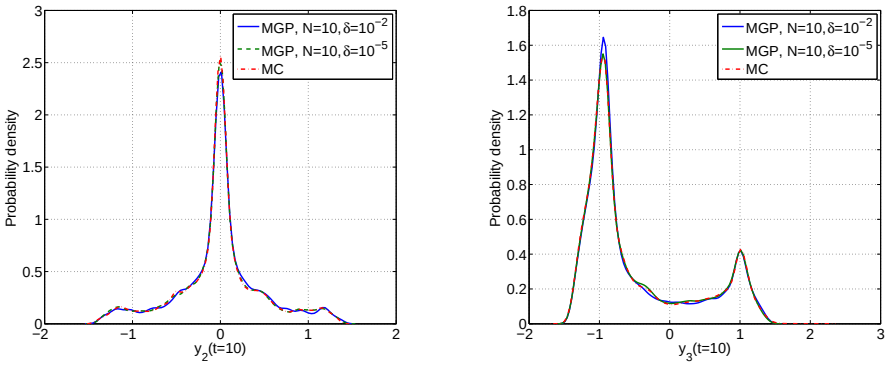


Figure 12: KO-2: predictive mean (dashed blue) versus MC estimate (solid red) of the mean (left column, $a,\ c,\ e$) and variance (right column, $b,\ d,\ f$) of $y_3(t)$ with 95% error bars for tolerances (top to bottom) $\delta = 10^{-4}, 10^{-6}$ and 10^{-8}.



(a) (b)

Figure 13: KO-2: kernel density estimation of the PDF of y_2 ($t = 10$) (left) and y_3 ($t = 10$) (right) using 10^5 samples.

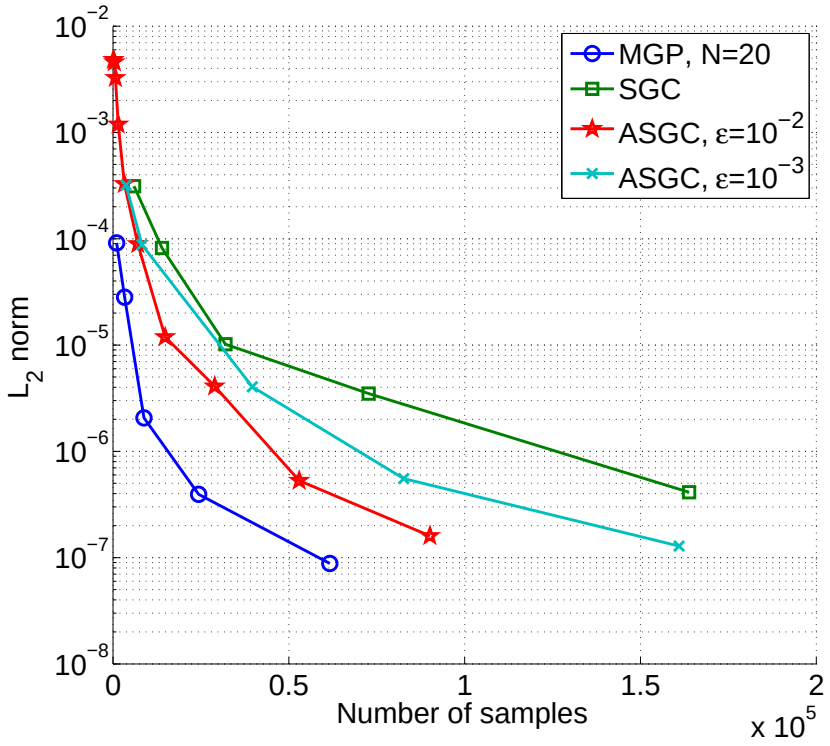


Figure 14: KO-3: the L_2 norm of the error in variance as a function of the observed samples for MGP, SGC and ASGC.

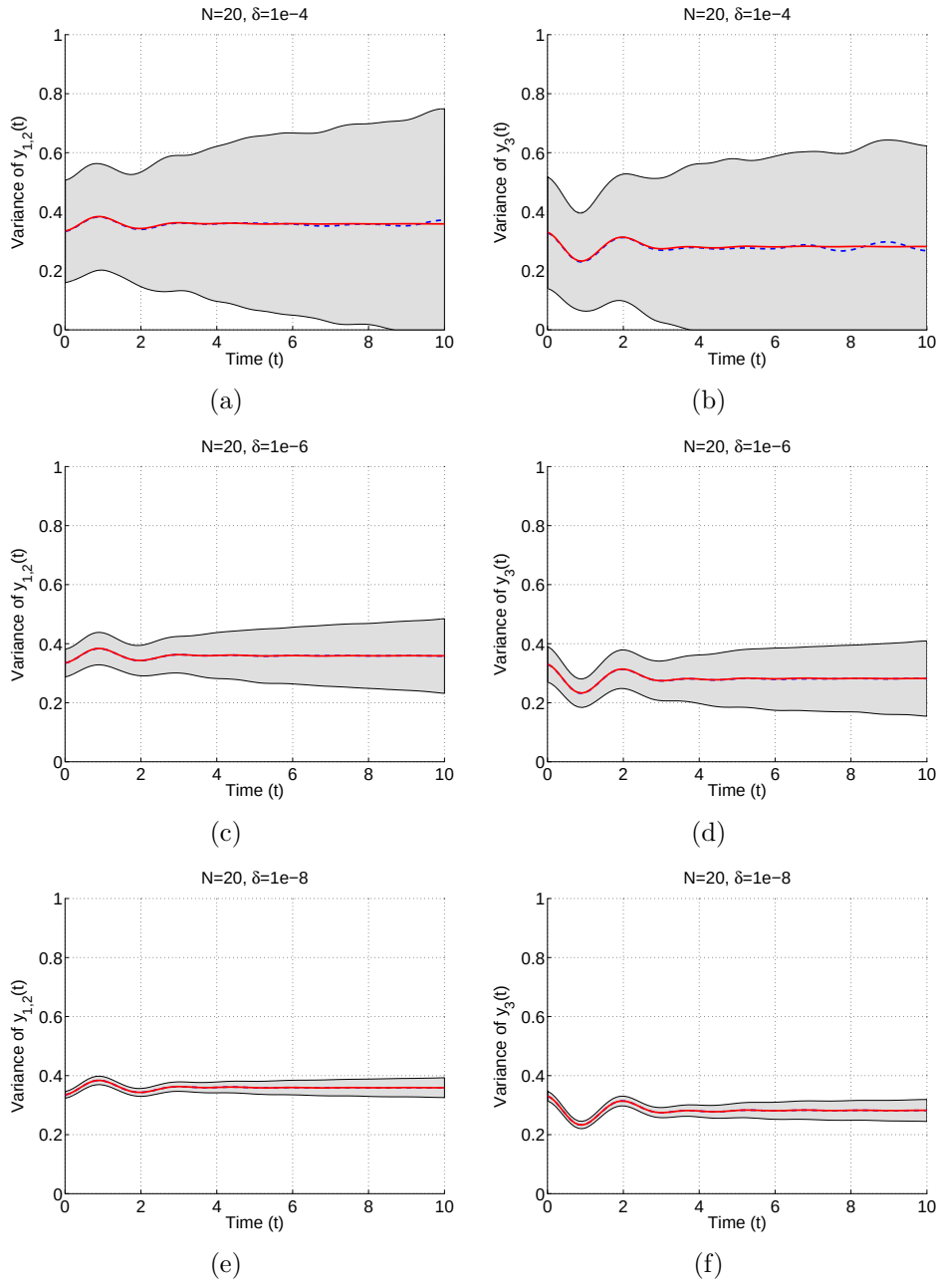
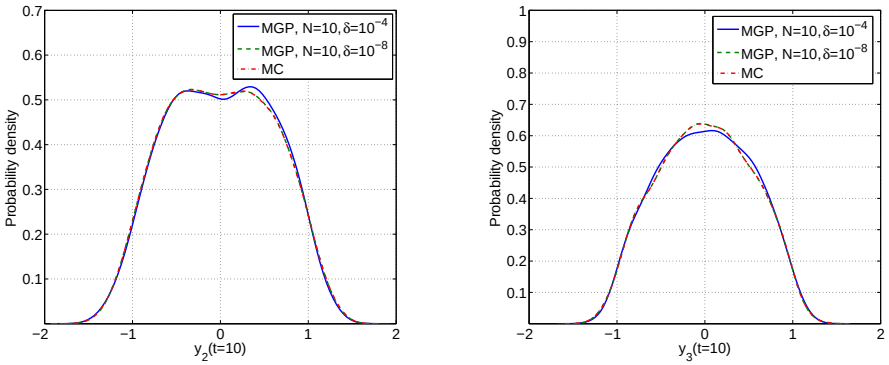


Figure 15: KO-3: predictive mean (dashed blue) versus the MC estimate (solid red) of the variance of $y_1(t)$ (left column, $a,\ c,\ e$) and $y_2(t)$ (right column, $b,\ d,\ f$) with 95% error bars for tolerances (top to bottom) $\delta = 10^{-4}, 10^{-6}$ and 10^{-8}.



(a) (b)

Figure 16: KO-3: kernel density estimation of the PDF of y_2 ($t = 10$) (left) and y_3 ($t = 10$) (right) using 10^5 samples.

3.3. Elliptic Problem

In this section, we consider a simple stochastic elliptic problem [41]. Consider the stochastic partial differential equation (SPDE):

$$\begin{aligned} -\nabla \cdot (a_K(\boldsymbol{\omega}, \cdot)\nabla u(\boldsymbol{\omega}, \cdot)) &= f(\cdot), \text{ in } D, \\ u(\boldsymbol{\omega}, \cdot) &= 0, \text{ on } \partial D, \end{aligned}$$

where the physical domain is $D = [0, 1]^2$. In order to avoid confusion with the physical dimension $\mathbf{x}$, we have chosen to denote the random variables with $\boldsymbol{\omega}$ instead of $\mathbf{x}$. We choose a smooth *deterministic* load:

$$f(x, y) = 100 \cos(x) \sin(y),$$

and work with homogeneous boundary conditions. The deterministic problem is solved with the finite element method using 400 (20×20 grid) bilinear quadrilateral elements. The random diffusion coefficient $a_K(\boldsymbol{\omega}, x)$ is constructed to have a one-dimensional dependence:

$$\log(a_K(\boldsymbol{\omega}, x, y) - 0.5) = 1 + \omega_1 \left(\frac{\sqrt{\pi}L}{2}\right)^{1/2} + \sum_{k=2}^{K} \xi_k \phi_k(x) \omega_k, \tag{55}$$

where

$$\xi_k := \left(\sqrt{\pi}L\right)^{1/2} \exp\left(\frac{-\left(\lfloor \frac{k}{2} \rfloor \pi L\right)^2}{8}\right), \text{ for } k \geq 2,$$

and

$$\phi_k(x) := \begin{cases} \sin\left(\frac{\lfloor \frac{k}{2} \rfloor \pi x}{L_p}\right) & \text{, if } k \text{ is even,} \\ \cos\left(\frac{\lfloor \frac{k}{2} \rfloor \pi x}{L_p}\right) & \text{, if } k \text{ is odd,} \end{cases}$$

$\lfloor \cdot \rfloor$ being the integer part of real number. We choose the $\omega_k, k = 1, \ldots, K$ to be independent identically distributed random variables:

$$\omega_k \sim U([-\sqrt{3}, \sqrt{3}]).$$

Hence, the stochastic input space is $\boldsymbol{\Omega} = [-\sqrt{3}, \sqrt{3}]^K$. Finally, we set:

$$L_p = \max\{1, 2L_c\} \text{ and } L = \frac{L_c}{L_p},$$

where L_c is called the *correlation length*. The expansion Eq. (55) resembles the Karhunen-Loève expansion of a two-dimensional random field with stationary covariance

$$\mathrm{Cov}[\log(a_K - 0.5)]((x_1, y_1), (x_2, y_2)) = \exp\left(-\frac{(x_1 - x_2)^2}{L_c^2}\right).$$

In this study, we set the correlation length to $L_c = 0.6$ and test the convergence of our method for $K = 10, 20$ and 40 input dimensions. The results for $K = 10, 20$ and 40 are evaluated by calculating the L_2 error in variance (Eq. (54)) using a plain MC estimate with 10^6 samples. The performance is compared to ASGC for various ϵ. The $K = 10,\ 20$ and 40 cases are solved using $N = 20,\ 40$ and 80 up to a tolerance of 10^{-7}, 10^{-5} and 10^{-4}, respectively. Figures 17, 18 and 19 show the L_2 error in variance for each case. In all cases MGP outperforms ASGC, especially when the number of samples is small. The error curves in Fig. 18 become asymptotically flat for all methods (MGP and ASGC) as a result of the MC accuracy being reached. Fig. 20 shows the convergence of the prediction for the variance of MGP as the tolerance threshold is lowered to $\delta = 10^{-7}$. Subfigure (e) of the same figure, plots the uncertainty of the variance $\sigma^2_{v_r}$ (Eq. (33)) at that tolerance. As already observed in previous examples, $\sigma^2_{v_r}$ over-estimates the true error. Fig. 21 tests the predictive capabilities of MGP for $K = 10$ at a tolerance $\delta = 10^{-6}$ on a random input point. We notice a good agreement with the true response.

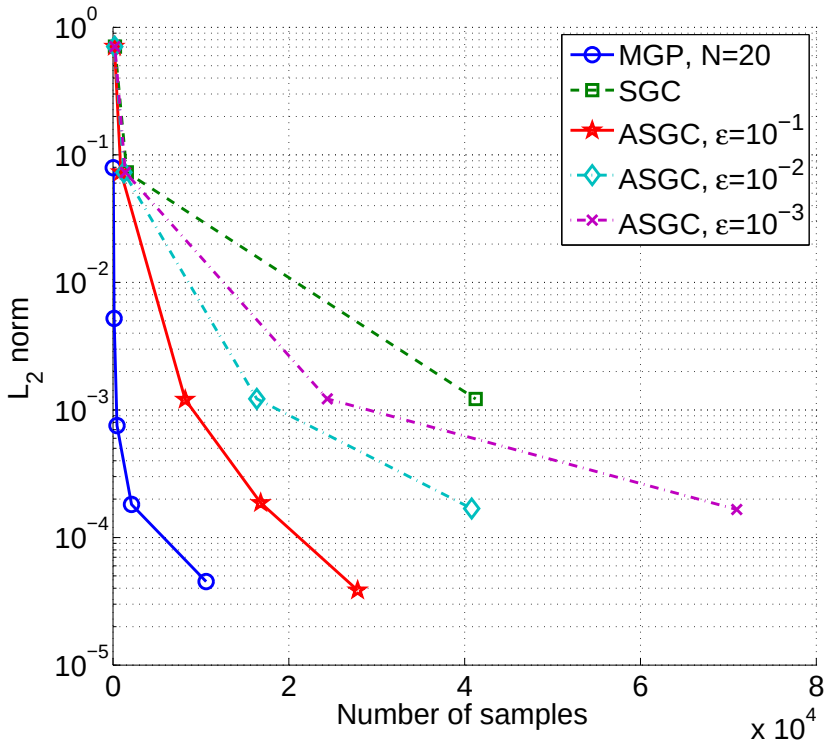


Figure 17: Elliptic, $K = 10$: The L_2 norm of the error in variance of the elliptic problem with $K = 10$ inputs as a function of the observed samples for MGP, SGC and ASGC.

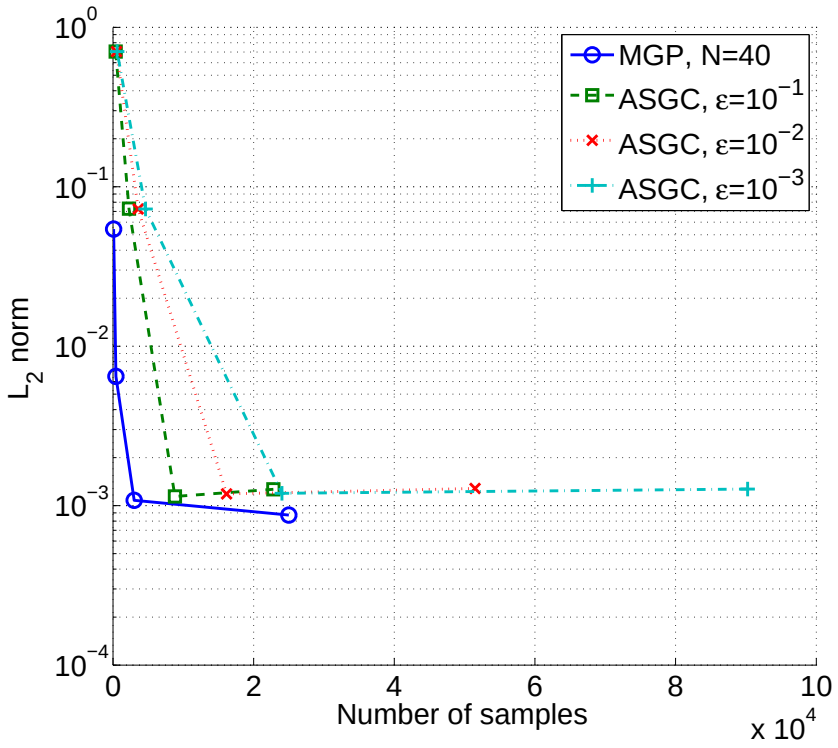


Figure 18: Elliptic, $K = 20$: The L_2 norm of the error in variance of the elliptic problem with $K = 20$ inputs as a function of the observed samples for MGP and ASGC.

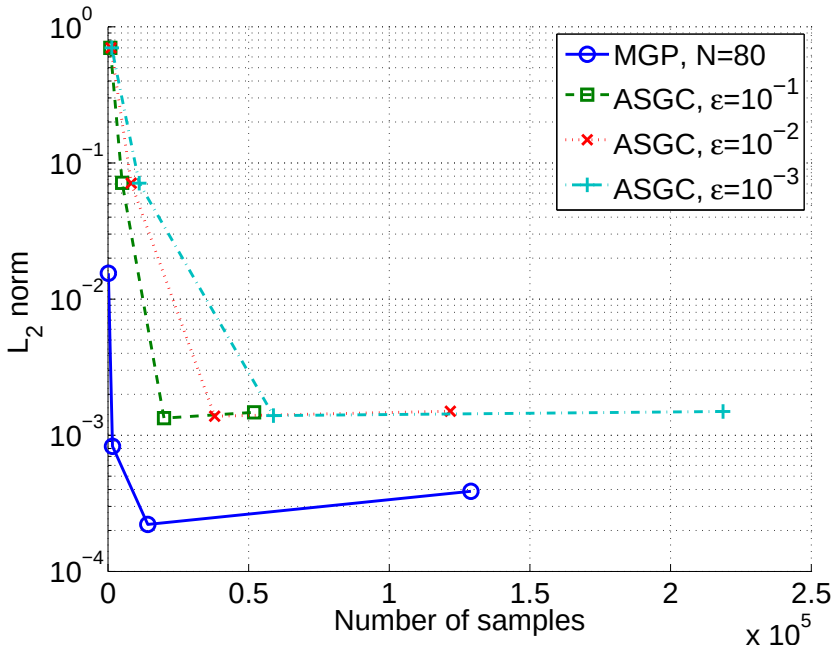


Figure 19: Elliptic, $K = 40$: The L_2 norm of the error in variance of the elliptic problem with $K = 40$ inputs as a function of the observed samples for MGP and ASGC.

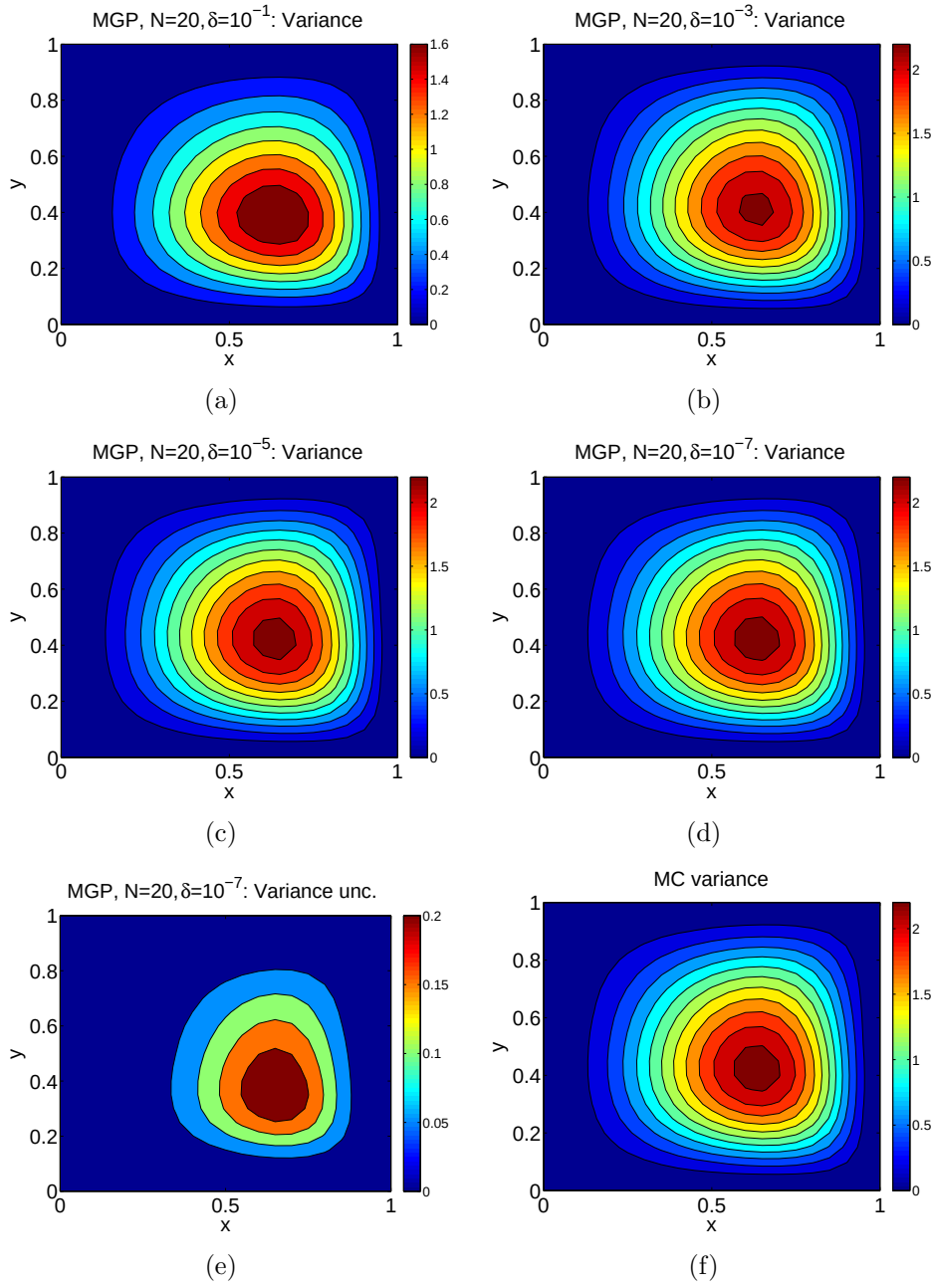


Figure 20: Elliptic, $K = 10$: Convergence of the predicted variance as the tolerance decreases. Subfigure (f) refers to MC results and subfigure (e) shows the uncertainty associated with the predicted variance $\sigma^2_{v_r}$ at $\delta = 10^{-7}$.

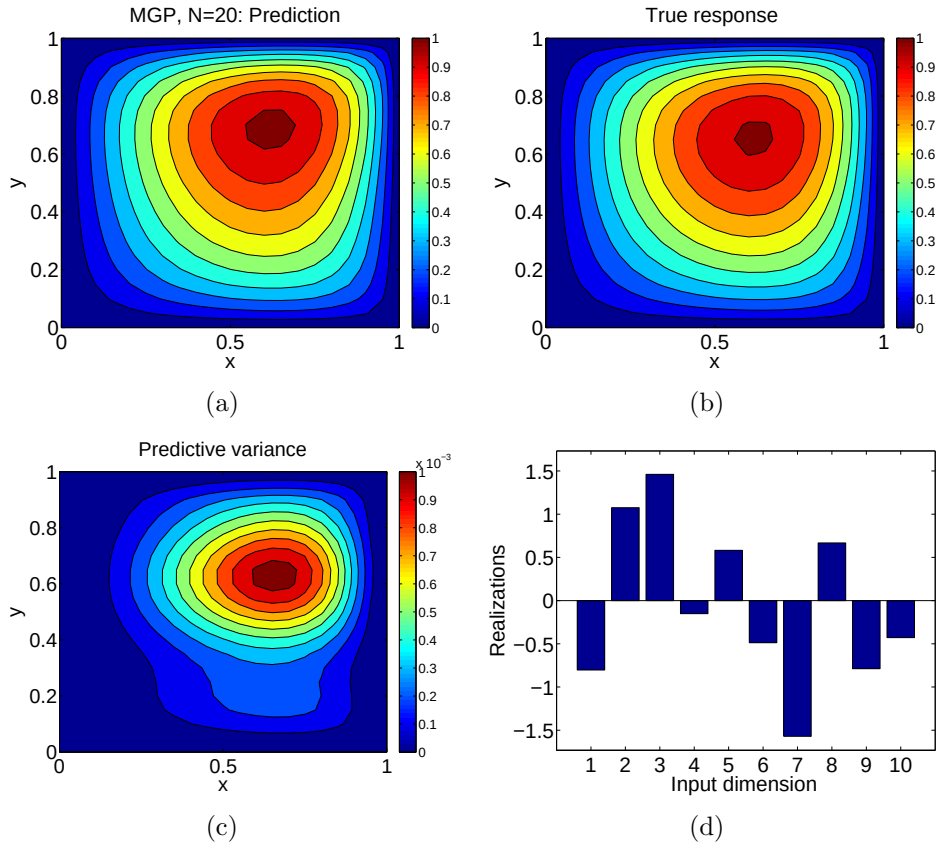


Figure 21: Elliptic, $K = 10, \delta = 10^{-6}$: Comparing the prediction (a) at a random input point (d) with the true response (b). Subfigure (c) shows the corresponding predictive variance.

3.4. Natural Convection Problem

Consider the dimensionless form of the Oberbeck-Boussinesq approximation using the vorticity transport equation in stream-function formulation:

$$\begin{aligned} -\frac{\partial}{\partial t}\nabla^2\psi - \frac{\partial\psi}{\partial y}\frac{\partial}{\partial x}\nabla^2\psi + \frac{\partial\psi}{\partial x}\frac{\partial}{\partial y}\nabla^2\psi &= -\operatorname{Pr}\nabla^4\psi + \operatorname{Ra}\operatorname{Pr}\frac{\partial T}{\partial x}, \\ \frac{\partial T}{\partial t} + \frac{\partial\psi}{\partial y}\frac{\partial T}{\partial x} - \frac{\partial\psi}{\partial x}\frac{\partial T}{\partial y} &= \nabla^2 T, \end{aligned}$$

where Pr and Ra are the Prandtl and Rayleigh numbers, respectively. In this formulation, the velocity field is given by:

$$u = \frac{\partial\psi}{\partial y}, \quad v = -\frac{\partial\psi}{\partial x}. \tag{56}$$

We solve the problem in a two-dimensional square cavity $\mathbf{X} = [0,1]^2$. We impose no slip conditions to the boundary:

$$u(x,y) = 0, \quad v(x,y) = 0, \text{ for } (x,y) \in \partial\mathbf{X}.$$

The two horizontal walls are considered adiabatic:

$$\frac{\partial T(x,y)}{\partial y} = 0, \text{ for } 0 \le x \le 1, y = 0, 1.$$

The right vertical wall (hot) is kept at a constant temperature:

$$T(1,y) = 0.5, \text{ for } 0 \le y \le 1.$$

The left vertical wall (cold) is taken to be a one-dimensional Gaussian stochastic process with mean -0.5 and exponential covariance

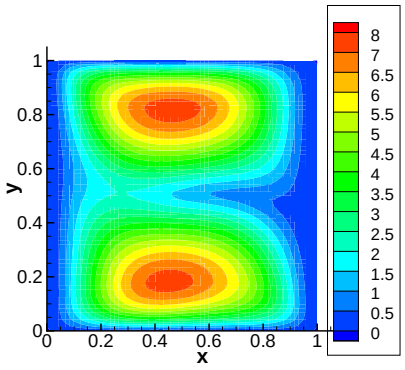
$$\operatorname{Cov}[x_1, x_2] = s^2 \exp\left\{-\frac{|x_1 - x_2|}{L_C}\right\},$$

where s^2 is the variance of the signal and L_C the correlation length. Using the Karhunen-Loève (KL) expansion, we may write

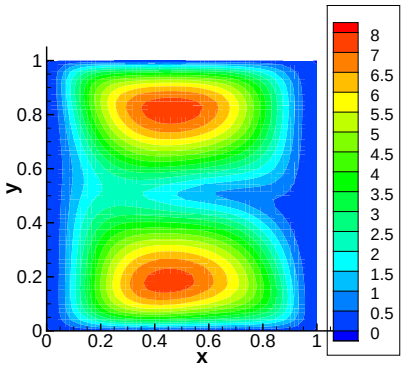
$$T(0,y;\boldsymbol{\omega}) = -0.5 + \sum_{k=1}^{\infty}\sqrt{\lambda_k}\phi_k(y)F^{-1}(\omega_k),$$

where λ_k and $\phi_k(y)$ are the eigenvalues and eigenvectors of the covariance function and F^{-1} is the inverse cumulative distribution function of $\mathcal{N}(0,1)$ and ω_k are independent uniform random variables in $[0,1]$. It is noted here that λ_k and $\phi_k(y)$ are analytically available [42].

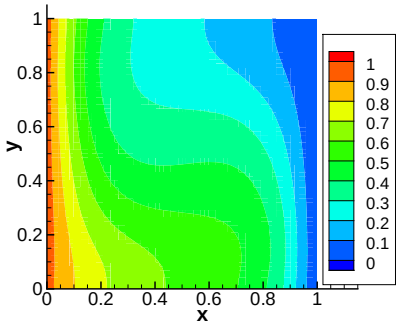
In this study, we set $L_C = 1$ and keep only $K = 4$ or 8 terms in the KL expansion. The parameters we use are $\mathrm{Pr} = 1$ and $\mathrm{Ra} = 5000$. The deterministic problem is solved using the Nektar fluid dynamics code [43], which utilizes spectral elements. The domain was decomposed in 240 quadrilateral elements (12×12 grid) and 4 spectral modes were used on each one. It has been numerically verified that no more modes were necessary for convergence of the spectral elements. The output is observed at 16 (4×4 grid) equidistant mesh points on each element. This results in a total of 2401 outputs for each of the physical quantities of interest (T, u, v and the pressure p). The total number of output dimensions is thus $M = 9604$. For computational convenience, we only work with temperature T and the u component of the velocity, a total of 4802 output parameters. For $K = 2$ and 4, we run our scheme until a tolerance $\delta = 10^{-5}$ is reached with $N = 10$. A total of 1393 and 14396 observations were made, respectively. For $K = 8$, we reach a tolerance of $\delta = 10^{-3}$ which results in 829 observations being made. Fig. 22 compares the predicted standard deviations (std.) of u (top) and T (bottom) for $K = 8$ with MC estimates using $80,000$ samples. The results are in good agreement with the MC estimates. In Figs. 23, we draw a random sample from the input distribution for the $K = 4$ case. We present the predictive mean of T along with two std.'s and compare it to the absolute error. Notice that the two std.'s are qualitatively similar to the absolute error of the prediction.



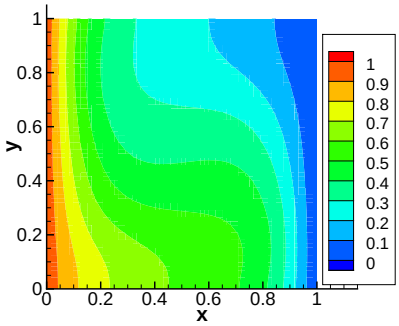
(a) K=8, MGP ($N = 20, \delta = 10^{-3}$): std. of u



(b) K=8, MC: std. of u

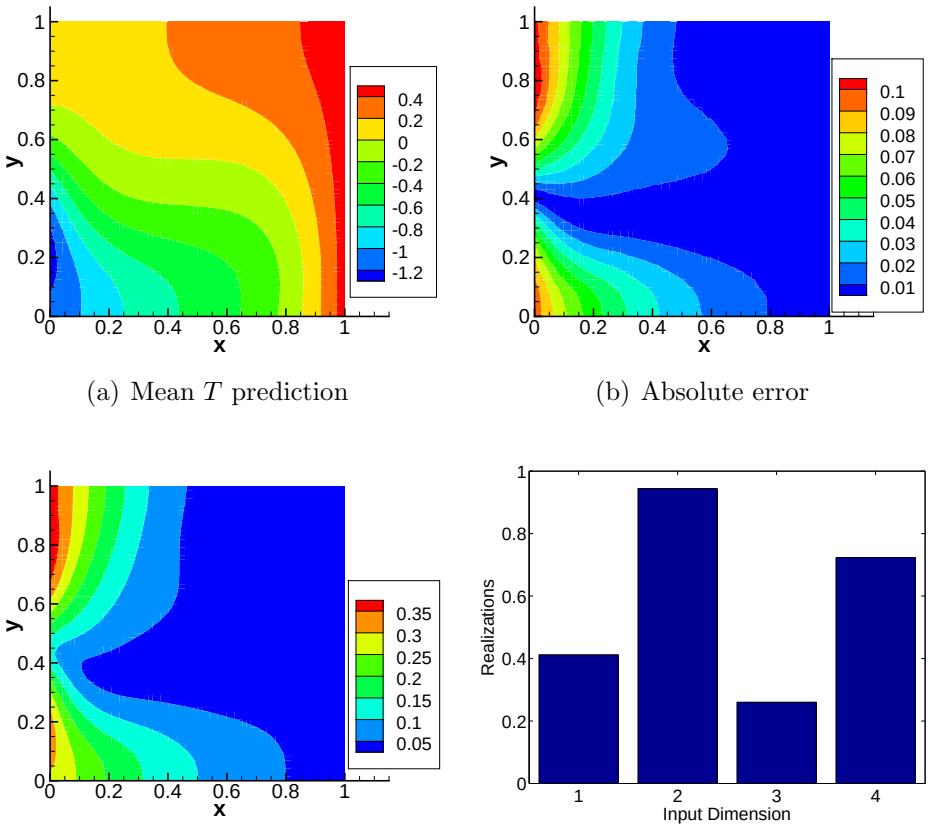


(c) K=8, MGP ($N = 20, \delta = 10^{-3}$): std. of T



(d) K=8, MC: std. of T

Figure 22: Natural Convection: MGP prediction at tolerance level $\delta = 10^{-3}$ for the standard deviation of the velocity u (top) and temperature T (bottom) compared to a MC estimate for $K = 8$ input dimensions.



(a) Mean T prediction

(b) Absolute error

(c) 2 std.'s of T

(d) Input point

Figure 23: Natural Convection ($K = 4, \delta = 10^{-5}$): Comparing the prediction at a random input point with the true response.

4. Conclusions

We have developed a novel, non-intrusive Bayesian scheme based on a treed multi-output GP model that can be used in UQ tasks. The tree is built in a sequential way that utilizes information contained only in the data observed so far. Tree refinement depends on the observations through a global measure of the uncertainty in the prediction, the inferred length scales as well as the input probability distribution. A Sequential Experimental Design technique based on the predictive uncertainty was also used to adaptively select the most informative input points on each element. The final result is a non-stationary, predictive distribution for the response of the underlying system, that can be semi-analytically integrated to provide point estimates and error bars for the statistics of interest. We have numerically demonstrated that the framework can (1) capture non-stationary responses, (2) locate discontinuities, (3) identify localized features and (4) reduce the sampling frequency on unimportant input dimensions. The method was shown to outperform SGC and ASGC in almost all numerical examples investigated, especially when only a small number of observations were used.

The presented framework is particularly interesting, in that it can be extended in several ways that can improve its performance dramatically. From a technical point of view several aspects require further numerical investigation: e.g. the dependence of the result on the choice of the maximum number of samples per element N, the performance of the ALC experimental design technique instead of the ALM scheme used in the current work, the dependence of the final decomposition of the stochastic space on the refinement criterion Eq. (40) for $q \neq 1$ and so on. Another important development would be to replace the current multi-output GP model with a GP model that explicitly takes into account correlation between the outputs. Such an effort, is expected to reduce the number of samples required significantly. Currently, the GPs learnt on each element are dropped if the element is split in half. The result is that each element is treated independently and the response is not smooth along the element boundaries. Alternatively, another treed GP model can be formulated in which the children of a node would learn the residual of the response instead of the response itself. In such a way, the upper nodes of the tree would model coarse features of the response, while localized features would be resolved by the leaves of the tree. Finally, a great deal of effort must be put in mathematically working out the error bounds in the various statistics that result from the uncertainty of the prediction.

As already mentioned in Section 2.2, the proper Bayesian way to account for the uncertainty of the predicted statistics, would be via an MC procedure: we would sample a complete response surface from the full model, integrate it with respect to the input probability distribution and obtain a sample of the statistics. The mathematical details of such a procedure are the subject of our current research.

Appendix A. Implementation Details

In this appendix, we discuss several details with regards to the implementation of the UQ framework presented.

The nugget. The covariance function we use has the special form:

$$c(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}; \boldsymbol{\theta}, g) = c(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}; \boldsymbol{\theta}) + g^2 \delta_{nm}, \tag{A.1}$$

where $c(\cdot, \cdot; \boldsymbol{\theta})$ is a normal covariance function depending on some hyper-parameters $\boldsymbol{\theta}$, $g^2 > 0$ and δ_{nm} is the Kronecker delta. Such a covariance function corresponds to the case where $\mathbf{f}(\mathbf{x})$ is observed with additive Gaussian noise with zero mean and variance g^2 (see p. 16 of [22]). In the literature of analysis of computer experiments using GPs, g^2 is known as the *nugget*. Many authors (e.g. [14], [15]), omit the nugget on the grounds that computer codes are deterministic. Inclusion of the nugget, however, has been observed to enhance numerical stability in factorizing the covariance matrix [44, 17]. On our part, we have observed that numerical stability is further improved, if a zero mean, g^2 Gaussian noise is added to the scaled observed responses Eq. (11). The effect of the nugget is the addition of a g^2 term in the predictive variance of the scaled responses. A typical value of the nugget we use in the numerical examples is $g^2 = 10^{-6}$. For a very recent discussion on the importance of the nugget in computer modeling, see [45].

Maximizing the marginal likelihood. In this work, we make exclusive use of the SE covariance function defined in Eq. (41). Its hyper-parameters are the signal strength $s_f > 0$ and the length scale of each stochastic input $\ell_k > 0$. Each stochastic element is associated with its local hyper-parameters which are found by maximizing the joint marginal likelihood subject to the positivity constraint. In order to achieve this in practice, we maximize with respect to the logarithm of these quantities, i.e. we re-parameterize the covariance function as:

$$\theta_1 = \log s_f, \ \theta_{k+1} = \log \ell_k.$$

This results in an equivalent unconstrained optimization problem which we solve using a Conjugate Gradient (CG) method [30], i.e. Eq. (12) with $\boldsymbol{\Theta} = \mathbb{R}^{K+1}$. It is important to notice that the nugget, g^2, is not optimized. It remains fixed to a given small value. Specifically, we used the Fletcher-Reeves CG algorithm [46] as implemented in GSL [40]. The starting values $\boldsymbol{\theta}_0 = (\theta_{1,0}, \ldots, \theta_{K+1,0})$ of the optimization algorithm are chosen as follows:

1. If we fit a GP for the first time (i.e. using $\mathbf{X}$ itself as the first element), we set $\theta_{1,0} = 0$ for the signal parameter and

$$\theta_{k+1,0} = \log\left(\frac{1}{3}L_k\right), \ k = 1, \ldots, K,$$

   for the length scale parameters, where $L_k = b_k - a_k$ is the extent of $\mathbf{X}$ along the k-dimension (Eq. (42)).
2. Otherwise, if $\mathbf{X}^i$ comes from splitting in half a parent element, we set $\boldsymbol{\theta}_0$ equal to the hyper-parameters of the parent element.

The optimization problem does not necessarily have a unique maximum. In reality, different local maxima are associated with different interpretations of the observed data set (Ch. 5 of [22]). In our numerical examples, we did not encounter any problems with this optimization and the maxima we obtained were quite robust. Powers of the response function are also treated as MGPs with SE covariance function, albeit having their own hyper-parameters $\boldsymbol{\theta}^q$ (see Section 2.2). These are also selected by maximizing the marginal likelihood.

Evaluation of the integrals. Finally, we come to the problem of computing the necessary integrals for the evaluation of the statistics (Eqs. (24), (25) and (26)). It is apparent that for general elements, input probability distribution and covariance function, these integrals have to be numerically evaluated. We choose to work with square elements, uniform input probability distribution and SE covariance function. With this choice, it is possible to express those integrals analytically using the error function:

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \tag{A.2}$$

In particular, let $\mathbf{X}^i = \times_{k=1}^K [a_k^i, b_k^i]$ and $p^i(\mathbf{z})$ be the uniform distribution on $\mathbf{X}^i$, i.e.

$$p^i(\mathbf{z}) = \frac{1_{\mathbf{X}^i}(\mathbf{z})}{\prod_{k=1}^K (b_k^i - a_k^i)}. \tag{A.3}$$

Then, it is easy to show that (for $q = 1$):

$$\epsilon^1(\mathbf{x}) = s_f^2 \left(\frac{\pi}{2}\right)^{K/2} \prod_{k=1}^{K} \ell_k^i \left(\Phi\left(\frac{b_k^i - x_k}{\sqrt{2}\ell_k^i}\right) - \Phi\left(\frac{a_k^i - x_k}{\sqrt{2}\ell_k^i}\right)\right) \tag{A.4}$$

and

$$\nu^1(\mathbf{x},\mathbf{y}) = \left(\frac{\pi}{2}\right)^{K/2} s_f^3 \sqrt{c(\mathbf{x},\mathbf{y})} \prod_{k=1}^{K} \ell_k^i \left(\Phi\left(\frac{2b_k^i - x_k - y_k}{2\ell_k^i}\right) - \Phi\left(\frac{2a_k^i - x_k - y_k}{2\ell_k^i}\right)\right). \tag{A.5}$$

The constant c^1 (Eq. (25)), can be trivially shown to be

$$c^1 = s_f^2. \tag{A.6}$$

The integrals that pertain to the higher moments $q > 1$ are obtained similarly by replacing the hyper-parameters with the ones that correspond to the MGP representing the response raised to the q power $\mathbf{f}^q$.

Acknowledgements

This research was supported by an OSD/AFOSR MURI09 award on uncertainty quantification, the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and the Computational Mathematics program of the National Science Foundation (NSF) (award DMS-0809062). This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Additional computing resources were provided by the NSF through TeraGrid resources provided by NCSA under grant number TG-DMS090007.

References

[1] R. L. Iman, W. J. Conover, Small sample sensitivity analysis techniques for computer models, with an application to risk assessment, Communications in Statistics - Theory and Methods 9 (17) (1980) 1749–1842.

[2] M. B. Giles, Multilevel Monte Carlo path simulation, Tech. rep., Oxford University Computing Laboratory (2006).

[3] M. B. Giles, Improved multilevel Monte Carlo convergence using the Milstein scheme, in: A. Keller, S. Heinrich, H. Niederreiter (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer, 2008, pp. 343–358.

[4] R. G. Ghanem, P. D. Spanos, Stochastic Finite Elements: A Spectral Approach, Dover Publications, 2003.

[5] D. Xiu, G. E. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, Journal of Scientific Computing (24) (2002) 619–644.

[6] X. Wan, G. E. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, Journal of Computaional Physics 209 (2005) 617–642.

[7] X. Wan, G. E. Karniadakis, Multi-element generalized polynomial chaos for arbitrary probability measures, SIAM Journall of Scientific Computing 28 (3) (2006) 901–928.

[8] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM Journal of Numerical Analysis 45 (3) (2007) 1005–1034.

[9] S. A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, in: Dokl. Akad. Nauk SSSR, Vol. 4, 1963, p. 123.

[10] D. Xiu, J. S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM Journal on Scientific Computing 27 (3) (2005) 1118–1139.

[11] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Communications in Computational Physics 2 (2) (2007) 293–309.

[12] F. Nobile, R. Tempone, C. G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, SIAM Journal of Numerical Analysis 46 (5) (2008) 2309–2345.

[13] X. Ma, N. Zabaras, An adaptive hierarchical sparse Grid Collocation algorithm for the solution of stochastic differential equations, Journal of Computaional Physics 228 (8) (2009) 3084–3113.

[14] J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn, Design and analysis of computer experiments, Statistical Science 4 (4) (1989) 409–435.

[15] T. J. Santer, B. J. Williams, W. I. Notz, The Design and Analysis of Computer Experiments, Springer, New York, 2003.

[16] D. J. C. MacKay, Information-based objective functions for active data selection., Neural Computation 4 (4) (1992) 590–604.

[17] R. B. Gramacy, H. K. H. Lee, Bayesian treed Gaussian Process models with an application to computer modeling, Journal of the American Statistical Association 103 (483) (2008) 1119–1130.

[18] H. A. Chipman, E. I. George, R. E. McCulloch, Bayesian CART model search, Journal of the American Statistical Association 93 (443) (1998) 935–948.

[19] H. A. Chipman, E. I. George, R. E. McCulloch, Bayesian treed models, Machine Learning 48 (1) (2002) 299–320.

[20] M. A. Taddy, R. B. Gramacy, N. G. Polson, Dynamic trees for learning and dsign, Journal of the American Statistical Association 106 (493) (2011) 109–123.

[21] D. J. C. MacKay, Gaussian processes, Tutorial at Neural Information Processing Systems 10.

[22] C. E. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.

[23] N. A. C. Cressie, Statistics for Spatial Data, Vol. 4, Wiley, New York, 1991.

[24] P. Boyle, M. Frean, Dependent Gaussian processes, in: Neural Information Processing Systems 18, 2005, pp. 217–224.

[25] Y. W. Teh, M. Seeger, M. I. Jordan, Semiparametric latent factor models, in: R. G. Cowell, Z. Ghahramani (Eds.), 10th International Workshop on Artificial Intelligence and Statistics, Society for Artificial Intelligence and Statistics, 2005, pp. 333–340.

[26] C. A. Micchelli, M. Pontil, Kernels for multi-task learning, in: L. K. Saul, Y. Weiss, L. Bottou (Eds.), Advances in Neural Information Processing 17, MIT Press, 2005, pp. 921–928.

[27] D. Higdon, J. Gattiker, B. Williams, M. Rightley, Computer model calibration using high-dimensional output, Journal of the American Statistical Association 103 (482) (2008) 570–583.

[28] S. Conti, A. OHagan, Bayesian emulation of complex multi-output and dynamic computer models, Journal of Statistical Planning and Inference 140 (3) (2010) 640 – 651.

[29] D. J. C. MacKay, Bayesian interpolation, Neural Computation 4 (3) (1992) 415–447.

[30] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes: The Art of Scientific Computing, 3rd Edition, Cambridge University Press, 2007.

[31] J. Foo, X. Wan, G. E. Karniadakis, The Multi-Element probabilistic collocation method (ME-PCM): error analysis and applications, Journal of Computational Physics 227 (22) (2008) 9572–9595.

[32] M. Abramowitz, I. A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Vol. 55, Dover Publications, 1964.

[33] R. M. Neal, Bayesian Learning for Neural Networks, Springer, New York, 1996.

[34] C. K. I. Williams, C. E. Rasmussen, Gaussian processes for regression, in: D. S. Touretzky, M. C. Mozer, M. E. Hasselmo (Eds.), Advances in Neural Information Processing Systems 8, MIT Press, 1996, pp. 514–520.

[35] K. Chaloner, I. Verdinelli, Bayesian Experimental Design : A Review, Statistical Science 10 (3) (1995) 273–304.

[36] D. A. Cohn, Neural network exploration using optimal experiment design., Neural Networks 9 (6).

[37] S. Seo, M. Wallat, T. Graepel, K. Obermayer, Gaussian process regression: active data selection and test point rejection, in: International Joint Conference on Neural Networks, Vol. 3, IEEE Press, Los Alamitos, CA, 2000, pp. 241–246.

[38] R. B. Gramacy, H. K. H. Lee, Adaptive design and analysis of supercomputer experiments, Technometrics 51 (2) (2009) 130–145.

[39] M. A. Heroux, J. M. Willenbring, Trilinos Users Guide, Tech. rep., Sandia National Laboratories (2003).

[40] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, GNU Scientific Library Reference Manual, 2009.

[41] F. Nobile, R. Tempone, C. Webster, A sparse grid collocation method for elliptic partial differential equations with random input data, SIAM Journal of Numerical Analysis 45 (2008) 2309–2345.

[42] D. Xiu, Numerical Methods for Stochastic Computations: A Spectral Method Approach, Princeton University Press, 2010.

[43] Nektar, Suite of simulation codes, `www.cfm.brown.edu/people/tcew/nektar.html`.

[44] R. M. Neal, Monte Carlo implementation of Gaussian process models for bayesian regression and classification, Tech. Rep. 9702, Departement of Statistics, University of Toronto, Toronto (1997).

[45] R. B. Gramacy, H. K. H. Lee, Cases for the nugget in modeling computer experiments, Statistics and Computing (to appear).

[46] R. Fletcher, Practical Methods of Optimization, 2nd Edition, Wiley, 1987.